

Article

# Manifold Learning Co-Location Decision Tree for Remotely Sensed Imagery Classification

Guoqing Zhou <sup>1,2,3,\*</sup>, Rongting Zhang <sup>2,3,\*</sup> and Dianjun Zhang <sup>3</sup>

<sup>1</sup> Guangxi Key Laboratory for Spatial Information and Geomatics, Guilin University of Technology, Guilin 541004, China

<sup>2</sup> College of Precision Instrument and Opto-Electronic Engineering, Tianjin University, Tianjin 300072, China

<sup>3</sup> The Center for Remote Sensing, Tianjin University, Tianjin 300072, China; zhangdianjun123@163.com

\* Correspondence: glitezhou@yahoo.com (G.Z.); zrt65@tju.edu.cn (R.Z.); Tel.: +86-773-589-6073 (G.Z.)

Academic Editors: Norman Kerle, Soe Myint, Clement Atzberger and Prasad S. Thenkabail

Received: 8 April 2016; Accepted: 11 October 2016; Published: 19 October 2016

**Abstract:** Because traditional decision tree (DT) induction methods cannot efficiently take advantage of geospatial knowledge in the classification of remotely sensed imagery, several researchers have presented a co-location decision tree (CL-DT) method that combines the co-location technique with the traditional DT method. However, the CL-DT method only considers the Euclidean distance of neighborhood events, which cannot truly reflect the co-location relationship between instances for which there is a nonlinear distribution in a high-dimensional space. For this reason, this paper develops the theory and method for a maximum variance unfolding (MVU)-based CL-DT method (known as MVU-based CL-DT), which includes unfolding input data, unfolded distance calculations, MVU-based co-location rule generation, and MVU-based CL-DT generation. The proposed method has been validated by classifying remotely sensed imagery and is compared with four other types of methods, i.e., CL-DT, classification and regression tree (CART), random forests (RFs), and stacked auto-encoders (SAE), whose classification results are taken as “true values.” The experimental results demonstrate that: (1) the relative classification accuracies of the proposed method in three test areas are higher than CL-DT and CART, and are at the same level compared to RFs; and (2) the total number of nodes, the number of leaf nodes, and the number of levels are significantly decreased by the proposed method. The time taken for the data processing, decision tree generation, drawing of the tree, and generation of the rules are also shortened by the proposed method compared to CL-DT, CART, and RFs.

**Keywords:** decision tree; co-location; MVU; imagery classification

## 1. Introduction

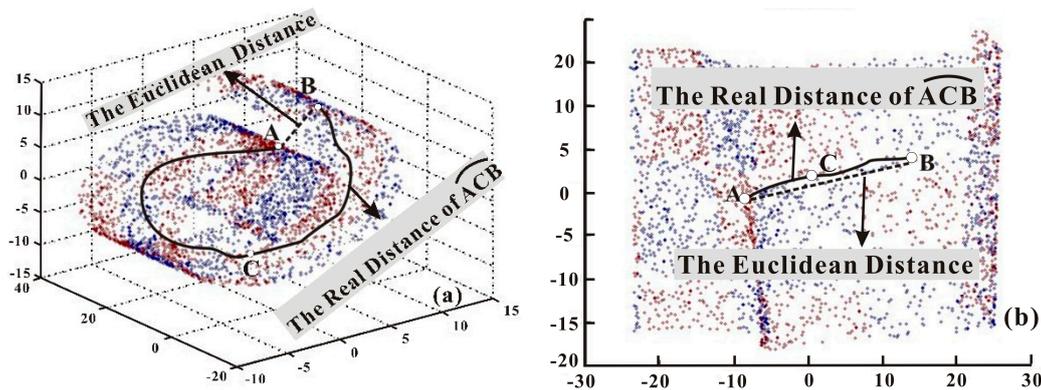
In the past several decades, a number of methods for classifying remotely sensed imagery have been proposed and improved [1–5]. Decision tree (DT) is one of the most prevalent methods [6–9]. The DT method’s application to remotely sensed image classification was first presented in 1975 by Wu et al. [10]. Since then, many researchers have endeavored to develop new theories and methods for constructing optimal DTs. The typical algorithms include, but are not limited to, classification and regression tree (CART) [11], iterative dichotomiser 3 (ID3) [12], C4.5 decision tree (C4.5) [13], and C5.0 decision tree (C5.0) [14]. Two primary issues in DT induction methods are: (1) the growth of the DT, where the split criteria are used to increase the number of nodes; and (2) the pruning phase of the DT, where the superfluous nodes and branches are removed to improve the classification accuracy. To construct a high-quality DT, many methods have been proposed in the literature, such as [15–18]. A number of methods for pruning have also been proposed in the literature, such as [19–24]. They have aimed to improve the classification accuracy by removing superfluous nodes and branches.

In recent years, many advanced/improved DT methods have been widely applied in image classification, especially for those areas with different objects but equivalent/similar spectral characteristics in the imagery. For example, Li et al. [25] combined the Dempster–Shafer (D-S) evidence theory with CART to propose a called D-S evidence theory DT method. This method can overcome the defect of traditional DTs in that they cannot address the uncertainty of remote sensing data and scene classification. This method can achieve very high accuracy in the classification of Landsat-5 TM image data. In addition, “hybrid” DT methods have been proposed by several researchers. For example, Farid et al. [26] proposed a “hybrid” DT algorithm that employs a naive Bayes classifier to remove the noisy troublesome instances from the training set before the DT induction. The experimental results demonstrated that the proposed method produced impressive results in the classification of challenging multi-class problems. In addition, to improve the classification accuracy, Feras Al-Obeidat et al. [27] proposed an improved decision tree algorithm based on a fuzzy method to classify Landsat satellite images that yielded better efficiency results and an interpretable model.

Traditional DT methods, however, only consider spectral information without efficiently taking advantage of the geospatial relationship between instances. Consequently, these methods unavoidably result in misclassifications, which are much worse in regions with different objects but equivalent/similar spectra. To overcome this type of shortcoming, Zhou et al. [19,28] presented a co-location-based decision tree (CL-DT) method, which has been successfully applied in road surface classification. Compared with the traditional DT, CL-DT can yield better classification results because the CL-DT method considers both geospatial attributes and non-geospatial attributes and employs co-location mining technology to first classify the co-location attributes of instances that have similar characteristics within a certain geospatial distance.

However, CL-DT only considers the Euclidean distance (ED), i.e.,  $distance(i, j)$ , as a geometric constraint condition. Although the ED can be used to effectively determine the co-location relationship between instances when they belong to a linear distribution in three-dimensional (3D) space, it is not capable of effectively determining their co-location relationship when they are a nonlinear distribution in 3D or higher-dimensional space [29]. For example, Figure 1a shows the data set of a Swiss roll in a 3D space, in which the ED  $\overline{AB}$  between instances A and B is shorter than the distance ACB, which is along the surface of the roll and effectively reflects whether there is a co-location relationship between A and B. This can be demonstrated by the fact that, if the Swiss roll is unfolded as in Figure 1b using the maximum variance unfolding (MVU) method based on the notion of local isometry, the MVU method can preserve the original neighborhood relationship of A and B, including the lengths of edges and the angles between edges at the same node [30]. After unfolding the input data, the unfolded distance (see Figure 1b) can effectively reflect the relationship between instances. Furthermore, in addition to the X/Y coordinates, the land surface temperature (LST), vegetation coverage (VC), and other factors are each able to be regarded as one of the dimensions of a high-dimensional space in which the instances belong to non-linear distributions.

For the problems mentioned above, this paper presents the maximum variance unfolding (MVU)-based co-location decision tree (MVU-based CL-DT) method. The contributions of this paper include: (1) an “unfolded distance” between instances is proposed for effectively reflecting the real co-location relationship between instances for which there is a nonlinear distribution in a high-dimension space; and (2) the proposed MVU-based CL-DT, induced by using the real co-location relationship between instances, can efficiently improve the classification accuracy of different objects with equivalent/similar spectral responds. In this paper, the classification results of SAE are considered as the “true value”. However, since SAE requires learning many parameters, the efficiency will be affected.



**Figure 1.** (a) Data of a Swiss roll; and (b) the result of mapping the Swiss roll data by MVU. (Note: Figure 1 is based on the original figure of Weinberger and Saul [31]).

## 2. MVU-Based Co-Location Decision Tree Induction Method

### 2.1. Brief Review of MVU

The MVU algorithm was first proposed by [30], whose goal was to detect and discover faithful low-dimensional structures in high-dimensional data [31]. The details of the MVU method can be found in [31]. Briefly, the MVU algorithm can faithfully preserve distances and angles among nearby input data instances by computing a low-dimensional representation of the high-dimensional data set [31]. After MVU processing, the neighboring instances of the input and output are invariant about the translation and rotation. Consequently, the unfolded distances between instances will be calculated to determine the co-location relationship between instances.

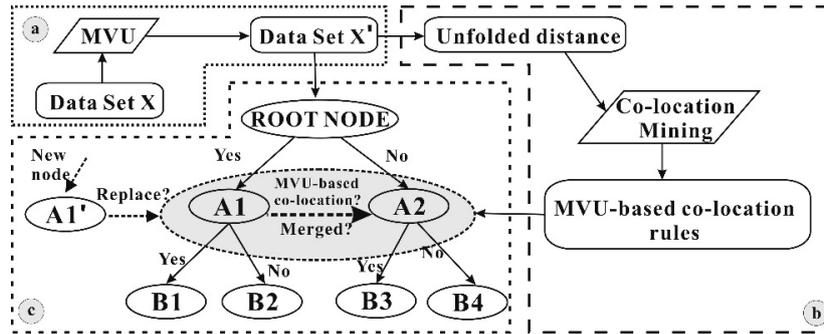
Over the past few decades, many efforts have been made to further develop the MVU method. Weinberger et al. [30,31] put forward an MVU algorithm that added a kernel matrix to the algorithm and used the positive semi definite constraint of the kernel matrix to realize the convex optimization of data. Shao et al. [32] discussed the deficiencies of kernel principal component analysis for monitoring nonlinear processes and proposed a new process monitoring technique based on maximum variance unfolding projections (MVUP). Liu et al. [33] analyzed the shortcomings of MVU for process monitoring and proposed an extended maximum variance unfolding (EMVU) method for nonlinear process monitoring. Ery et al. [34] analyzed and discussed the convergence of maximum variance unfolding, and they proved that it is consistent when the underlying sub manifold is isometric to a convex subset.

### 2.2. Basic Idea of the Proposed MVU-Based CL-DT

The details of the proposed method consist of three parts (see Figure 2).

1. Unfold the input data using the MVU method. To obtain the unfolded distance between instances, the original input dataset  $X$ , which is a nonlinear distribution in higher-dimensional space, is preprocessed using the MVU method. A new dataset  $X'$ , which is regarded as a linear distribution, is obtained and employed to: (1) calculate the unfolded distance; and (2) become the root node of the DT.
2. Mine co-location rules. Co-location rules are mined through a co-location algorithm with the unfolded distances, which are calculated from the dataset  $X'$ .
3. Generate MVU-based CL-DT. The dataset  $X'$  is taken as the input data of the root node of the DT. At the beginning, all attributes of dataset  $X'$  are “accepted” by the root node, and one “best” attribute, such as  $BA_1$ , is selected from the input dataset  $X'$ ; then, a splitting criterion is used to determine whether the root node will be split using a binary decision (Yes or No in Figure 2) into two intermediate nodes, noted by  $w_i$  and  $w_j$  ( $w = A, i = 1, \text{ and } j = 2$  in Figure 2). For each of the intermediate nodes ( $w_i$  and  $w_j$ ), the splitting criterion will be applied to determine whether the

node should be further split. If *No*, this node is considered a leaf node, and one of the class labels is assigned to this leaf node. If *Yes*, this node will be split by selecting another “best” attribute,  $BA_2$ . Meanwhile, the MVU-based co-location criterion will be used to judge whether the “best” attribute  $BA_2$  is co-located with  $BA_1$  at the same layer. If *Yes*, this node will be “merged” into the node and named a co-location node. For example, nodes A1 and A2 are merged into a new node A1' in Figure 2. After that, one new “best” attribute will again be selected, and whether the selected “best” attribute co-occurs with the last “best” attribute is re-judged. If *No*, the node will be further split into a sub-set by repeating the above operation. These above processes are repeated until all data have been processed.



**Figure 2.** Maximum variance unfolding (MVU)-based co-location decision tree (MVU-based CL-DT) induction method.

### 2.3. MVU-Based Co-Location Mining Rules

#### 2.3.1. MVU Unfolded Distance Algorithm

The MVU unfolded distance algorithm consists of the following three steps: (a) neighbor relation matrix reservation; (b) MVU function establishment and solution; and (c) calculation of the unfolded distance between instances. They are detailed as follows.

- Neighbor Relationship Matrix Reservation

The mapping function of an MVU method should preserve the neighborhood relationship of instances, including the angles and distances of instances. For this reason, the matrices  $X = \{\vec{X}_i\}_{i=1}^N$  and  $Y = \{\vec{Y}_i\}_{i=1}^N$  denote the input dataset and output dataset, respectively, which have an exactly one-to-one correspondence to the setup to preserve the neighborhood relationship. We define a matrix  $\delta$  to preserve the neighborhood relation among instances for  $X$  (and similarly,  $y$  for  $Y$ ). If the input instances  $X_i$  and  $X_k$  ( $i \neq k$ ) are  $k$ -nearest neighbors,  $\delta_{ik} = 1$ ; alternately, if  $X_i$  and  $X_k$  ( $i \neq k$ ) are not  $k$ -nearest neighbors but another input instance  $X_j$  is embedded between  $X_i$  and  $X_k$ , which makes  $X_i$  and  $X_k$   $k$ -nearest neighbors,  $\delta_{ik} = 1$  as well; otherwise,  $\delta_{ik} = 0$  (and similarly for  $Y_i$  and  $Y_k$ ).

- MVU Function Establishment and Solution

The MVU algorithm “unfolds” the input data by maximizing the sum of pairwise squared distances among the output data. Therefore, the objective function can be expressed by:

$$\begin{aligned}
 & \text{Max} \{ \sum_{i,k} \| \vec{Y}_i - \vec{Y}_k \|^2 \} \quad \text{subject to :} \\
 (1) & \quad \| \vec{Y}_i - \vec{Y}_k \|^2 = \| \vec{X}_i - \vec{X}_k \|^2 \quad \text{for all } (i,k) \text{ with } \delta_{ik} = 1 \\
 (2) & \quad \sum_i \vec{Y}_i = 0.
 \end{aligned}
 \tag{1}$$

where  $\|\vec{X}_i - \vec{X}_k\|^2$  and  $\|\vec{Y}_i - \vec{Y}_k\|^2$  are the squared distances of the input instances  $(\vec{X}_i, \vec{X}_k)$  and output instances  $(\vec{Y}_i, \vec{Y}_k)$ , respectively. Here, the distances between nearby inputs are enforced to match distances between nearby outputs by the first constraint. Meanwhile, a unique solution (up to rotation) is yielded by the second constraint, centering the outputs on the origin.

To finesse the apparent intractability of the above quadratic program, defining an inner product matrix  $L$  ( $L_{ik} = \vec{Y}_i \cdot \vec{Y}_k$ ), the objective function can be rewritten as follows (see [35,36] for details):

$$\begin{cases} \text{Max } \{tr(L)\} \\ \text{s.t. : } (1) \delta_{ik}(L_{ii} - 2L_{ik} + L_{kk}) = \delta_{ik}D_{ik} \\ (2) \sum_{ik} L_{ik} = 0 \\ (3) L \geq 0 \end{cases} \quad (2)$$

The third constraint requires the inner product matrix  $L$  to be positive semidefinite. After solving (see [35,36] for details), the inner product matrix  $L$  can be obtained, and then the output instance  $\vec{Y}_i$  can be obtained by utilizing the method of matrix diagonalization. This algorithm can be described as follows.

Let  $U_{\beta i}$  represent the  $\beta$ -th element of the  $i$ -th eigenvector, with eigenvalue  $\lambda_{\beta}$ ; then, the inner product matrix  $L$  can be rewritten as:

$$L_{ik} = \sum_{\beta=1}^n \lambda_{\beta} U_{\beta i} U_{\beta k} \quad (3)$$

Combining Equation (3) with  $L_{ik} = \vec{Y}_i \cdot \vec{Y}_k$ , the  $\beta$ -th element of the output instance  $\vec{Y}_i$  is expressed by:

$$\vec{Y}_{\beta i} = \sqrt{\lambda_{\beta}} U_{\beta i} \quad (4)$$

- Calculation of the Unfolded Distance between Instances

Based on the preserved neighbor relation matrix, we can use the established MVU model to unfold input data, and then the unfolded distances between instances are calculated as follows. If  $Y_i$  and  $Y_j$  are  $k$ -nearest neighbors, the unfolded distance, noted as  $D_U(Y_i, Y_j)$ , can be replaced by the Euclidean distance of  $Y_i$  and  $Y_j$ , noted as  $D_E(Y_i, Y_j)$ . Otherwise, if they are not  $k$ -nearest neighbors,  $D_U(Y_i, Y_j)$  is represented by the shortest cumulative Euclidean distance of a number of neighbors. This criterion can be mathematically described by:

$$D_U(Y_i, Y_j) = \begin{cases} D_E(Y_i, Y_j) & \text{if } Y_i \text{ and } Y_j \text{ are neighbor} \\ \min\{D_U(Y_i, Y_j), D_U(Y_i, Y_k) + D_U(Y_k, Y_h) + \dots + D_U(Y_g, Y_j)\} & \text{otherwise} \end{cases} \quad (5)$$

To further explain the unfolded distance between instances, an example is given in Figure 3. As observed from Figure 3,  $k$ -nearest neighbors are connected by lines, such as  $Y_1, Y_2$ , and  $Y_3$ . In Figure 3, the following two cases are considered: (1) the calculation of the unfolded distance between  $k$ -nearest neighbors; and (2) the calculation of the unfolded distance between instances that are not  $k$ -nearest neighbors. For  $k$ -nearest neighbors, such as  $Y_1$  and  $Y_3$ , the unfolded distance is  $D_E(Y_1, Y_3)$ . However, for  $Y_{10}$  and  $Y_1$ ,  $Y_{10}$  is not one of the  $k$ -nearest neighbors of  $Y_1$ . Hence, on the basis of Equation (5),  $D_U(Y_1, Y_{10})$  is the shortest cumulative Euclidean distance of a number of neighbors (such as  $Y_1, Y_3, Y_5, Y_6, Y_9$ , and  $Y_{10}$ ). Because the output data of MVU can be viewed as a connected graph and two arbitrary instances can be connected by a number of intermediate  $k$ -nearest neighbors, the shortest path can always be found.

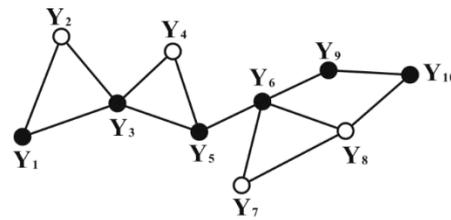


Figure 3. Determination of the unfolded distance between instances.

With the above analysis, the MVU unfolded distance (MUD) algorithm can be summarized as Algorithm 1, which calculates the unfolded distance after unfolding input data using MVU [30].

---

**Algorithm 1. The Algorithm of MUD**

---

1. **Input:** The number of nearest neighbors  $k$ ; Original data set  $X$ ; An  $N \times N$  zero matrix  $\delta$
  2. **Output:** Low-dimensional representation  $Y$ ; An  $N \times N$  binary matrix  $\delta'$ ; An Unfolded distance matrix  $U$
  3. **Process:**
  4. Step 1: Construct the neighbor graph.
  5. If  $X_i$  and  $X_k$  satisfy K-NN Then  $\delta'_{ik} = 1$ ; else  $\delta'_{ik} = 0$ ;
  6. Step 2: Semidefinite programming.
  7. Compute the maximum variance unfolding Gram matrix  $L$  of sample pairwise points, which is centered on the origin.
  8.  $\text{Max}\{\text{tr}(L)\}$  s.t.  $L \geq 0, \sum_{ik} L_{ik} = 0$  and  $\delta_{ik}(L_{ii} - 2L_{ik} + L_{kk}) = \delta_{ik}D_{ik}$ .
  9. Step 3: Compute low-dimensional embedding.
  10. Perform generalized eigenvalue decomposition for the Gram matrix  $L$  obtained by Step 2. The eigenvectors, corresponding to the front  $d$  greater eigenvalues, are the result of embedding.
  11.  $\vec{Y}_{\beta i} = \sqrt{\lambda_{\beta}} U_{\beta i}$
  12. Step 4: Calculate the unfolded distance of instances  $Y_i$  and  $Y_j$
  13. 
$$D_U(Y_i, Y_j) = \begin{cases} D_E(Y_i, Y_j) & \text{if } Y_i \text{ and } Y_j \text{ are neighbor} \\ \min\{D_U(Y_i, Y_j), D_U(Y_i, Y_k) + D_U(Y_k, Y_h) + \dots + D_U(Y_g, Y_j)\} & \text{otherwise} \end{cases}$$
- 

2.3.2. Determination of MVU-based Co-Locations

The successful completion of the above steps only obtains the unfolded distance between instances. With the unfolded distances, an R-relationship (RRS) between instances, which describes the neighborhood relationship between geospatial instances [6,27], can be created by the method proposed below.

If and only if the unfolded distance between  $Y_i$  and  $Y_j$  is less than or equal to the distance threshold, there exists an RRS between them, and it is mathematically expressed by:

$$R(Y_i, Y_j) = \begin{cases} 1 & \text{if } D_U(Y_i, Y_j) \leq D_{\theta} \\ 0 & \text{if } D_U(Y_i, Y_j) > D_{\theta} \end{cases} \quad (6)$$

where  $D_{\theta}$  is the threshold of unfolded distance,  $R(Y_i, Y_j)$  represents the RRS of  $Y_i$  and  $Y_j$ , and  $D_U(Y_i, Y_j)$  is the unfolded distance of  $Y_i$  and  $Y_j$ . After the determination of the RRS, those that satisfy the RRS condition are the candidates of MVU-based co-location.

After the process above, only the 2nd-order candidates of MVU-based co-location are obtained. To determine MVU-based co-locations, different attributes (e.g., SSM, LST, and VC) in the image classification of every instance are utilized. To this end, a density ratio ( $DR$ ) is defined to determine whether instances are MVU-based co-location patterns, and it is expressed by:

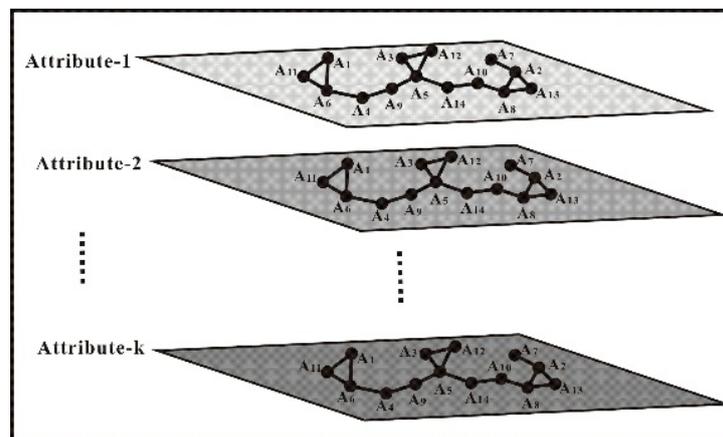
$$DR = \frac{\text{similar\_AttrN}(Y_i, Y_j)}{\text{Total\_AttrN}} \quad (7)$$

where  $Y_i$  is the  $i$ -th instance,  $similar AttrN(Y_i, Y_j)$  is the number of attributes for which the values of  $Y_i$  and  $Y_j$  satisfy the same threshold, and  $Total AttrN$  is the total number of attributes used in this paper. If  $DR$  is greater than or equal to the threshold  $DR_\theta$ ,  $(Y_i, Y_j)$  is a 2nd-order MVU-based co-location pattern.

When the 2nd-order MVU-based co-location patterns are determined as above, the  $k$  ( $k \geq 3$ )-order MVU-based co-location patterns should be constructed. The method for constructing the  $k$  ( $k \geq 3$ )-order MVU-based co-location patterns is as follows.

Let the  $(k-1)$ th ( $k \geq 3$ )-order MVU-based co-location pattern  $C_{k-1}$  connect the  $(k-2)$ th-order  $C_{k-2}$  to generate the  $k$ -th-order candidate of the MVU-based co-location pattern  $C'_k$ , and if  $C'_k$  satisfies the threshold of  $DR$ , it is regarded as the  $k$ -order MVU-based co-location pattern  $C_k$ . Take the constructed process of the 3rd-order MVU-based co-location  $(A_1, A_6, A_{11})$  in Figure 4 as an example. First,  $(A_1, A_6)$  connects  $(A_6, A_{11})$  using the previous  $k-2$  order, and then, the candidate of the MVU-based co-location pattern  $(A_1, A_6, A_{11})$  is obtained. If the  $DR$  of  $(A_1, A_6, A_{11})$  satisfies the threshold  $DR_\theta$ ,  $(A_1, A_6, A_{11})$  is an MVU-based co-location pattern. By parity of reasoning, all MVU-based co-location patterns are obtained.

Because those instances with MVU-based co-location patterns have similar characteristics, the phenomenon of different objects having the same spectrum in remote sensing images will be eliminated. Thus, the classification accuracy can be improved.



**Figure 4.** Determination of MVU-based co-locations (Note: Attribute- $k$  represents the  $k$ -th attribute, and  $A_i$  represents the  $i$ -th instance of object A. In each of the attributes, the location of instance  $A_i$  is the same).

### 2.3.3. Determination of Distinct Event-Types

These above processes only mine the MVU-based co-location instances and cannot ensure that these instances are distinct event types. Therefore, a constraint condition must be set up as below.

Let  $M = \{m_1, m_2, \dots, m_c\}$  be a set of corresponding cluster centers of attributes  $a_1, a_2, \dots, a_c$  such that the distinct event-type can be expressed by:

$$\Psi_i = \sum_{i=1}^S \sum_{k=1}^C (\|f_i - m_k\|)^2 \tag{8}$$

where  $\|f_i - m_k\|$  denotes the Euclidean distance between  $f_i$  and  $m_k$ ;  $f_i$  is the value of the  $i$ -th instance in the  $k$ -th attribute  $a_k$  of one class;  $m_k = \sum_{i=1}^N \frac{f_i}{N}$  is the cluster center of the  $k$ -th attribute  $a_k$  of one class, where  $N$  is the number of instance of one class;  $\Psi_i$  is a squared error clustering criterion;  $C$  is the number of attributes; and  $S$  is the number of instances. Therefore, if  $\Psi_i$  is greater than a given threshold  $\Psi_\theta$ , the  $i$ -th instance is regarded as a distinct event.

### 2.3.4. Generation of MVU-Based Co-Location Rules

Accompanying the generation of MVU-based co-locations, the MVU-based co-location rules can be created from the candidates of MVU-based co-location. With the methods proposed above, the algorithm and pseudo code for determination of MVU-based co-locations (MCL) are developed in this paper and can be described via Algorithm 2, which uses unfolded distance to replace Euclidean distance of [19].

---

#### Algorithm 2. the Algorithm of the determination of MCL

---

1. **Input:** Unfolded distance matrix  $D_U$ ; Unfolded dataset  $Y$ ; Thresholds of distance  $D_\theta$ , and thresholds of density radio  $\mathfrak{R}_\theta$ ;
  2. **Valuable:**  $\zeta$ : the order of a co-location pattern;  $R$ : the set of R-relationship between instances;  $C_\zeta$ : the set of candidates of a co-location pattern whose size is  $\zeta$ ;  $Tab_\zeta$ : the set of co-location patterns;  $Rul_\zeta$ : the  $\zeta$ -order co-location rule.
  3. **Output:**  $\zeta$ -order co-location pattern;  $Rul_\zeta$ .
  4. **Process:**
  5. Step 1:
  6. If  $(D_U(Y_i, Y_j) \leq D_\theta)$ , then  $(R(Y_i, Y_j) = 1)$  else  $(R(Y_i, Y_j) = NAN)$ ;
  7. Step 2:
  8. MVU\_co-location size  $\zeta = 1$ ;  $C_1 = Y_i$ ;  $P_1 = Y_i$ ;
  9. Step 3:
  10.  $Tab_1 = \text{generate\_table\_instance}(C_1, P_1)$ ;
  11. Step 4:
  12. If  $(\text{fmul} = \text{TRUE})$  then  $Tab_{C_1} = \text{generate\_table\_instance}(C_1, \text{multi\_event})$ ;
  13. Step 5:
  14.  $R = \text{determine\_R-relationship}(D_U(Y_i, Y_j))$ ;
  15. While  $(P_\zeta$  is not empty and  $\zeta < k)$  do
  16.  $\{C_\zeta + 1 = \text{generate\_candidate\_MVU-co-location}(C_\zeta, \zeta)$
  17.  $Tab_{C_\zeta + 1} = \text{generate\_MVU-co-location}(\mathfrak{R}_\theta, C_\zeta, \mathfrak{R})$
  18.  $Rul_\zeta + 1 = \text{generate\_MVU-co-location\_rule}(Tab_{C_\zeta + 1}, \beta)$ ;  $\zeta++$  }
  19. Return  $\zeta$ -order co-location pattern;  $Rul_\zeta$ .
- 

### 2.4. Pruning

In a DT induction algorithm, different pruning algorithms are employed to prune the nodes. In this paper, a pruning algorithm proposed by Zhou et al. [19] is employed. The pruning algorithm is briefly described as follows.

For a geospatial data set  $SP$ , let  $FT = (ft_1, ft_2, ft_3, \dots, ft_h)$  be a set of geospatial attributes. Let  $\Phi = (\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n)$  be a set of  $n$  instances in  $SP$ , where each instance is a vector containing the instance-id, geospatial attributes, and location. The geospatial attribute of instance  $i$  is denoted by  $ft_i$ . It is assumed that the geospatial attributes of an instance are from the geospatial attribute set  $FT$  such that the location is within the geospatial framework of a geospatial database and there is an RRS in  $SP$ . Additionally, let  $O = (o_1, o_2, o_3, \dots, o_k)$  be a set of corresponding clusters centered in the data set  $SP$ , where  $k$  is the number of clusters of geospatial attributes. To capture the concept of “nearby,” the criterion of co-occurrence is defined as:

$$\Xi_m = \sum_{i=1}^h \sum_{q=1}^n v_{iq} (\|\varphi_q - o_i\|)^2 \quad (9)$$

where  $\|\varphi_q - o_i\|$  is the Euclidean distance between  $x_q$  and  $o_i$ ;  $\Xi_m$  is the squared error clustering criterion; and  $V = \{v_{iq}\}$ ,  $i = 1, 2, \dots, h$ ,  $q = 1, 2, \dots, N$ , is a matrix that satisfies the following constraint conditions:

$$v_{iq} \in [0, 1], \quad \forall_i = 1, 2, \dots, h, \forall_q = 1, 2, \dots, N \quad (10)$$

$$\sum_i^h v_{iq} = 1, \quad \forall_i = 1, 2, \dots, h, \forall_q = 1, 2, \dots, N \quad (11)$$

Thus, if  $\Xi_m$  is less than or equal to a given threshold, the two nodes are considered to be a co-occurrence and thus should be merged.

### 2.5. Inducting Decision Rules

After the MVU-based CL-DT is generated above, decision rules will be created by translating the decision tree into semantic expressions. Because the MVU-based CL-DT algorithm partitions a data space into several distinct disjoint regions via axis-parallel surfaces, the top-down search method will be employed to translate individual nodes into rules in this paper.

As a summary, through introducing the MVU used to obtain unfolded distance into CL-DT [19], the entire proposed MVU-based CL-DT algorithm is described in Algorithm 3.

---

#### Algorithm 3. the Algorithm of MVU-based CL-DT

---

1. **Input:** Training dataset  $TD$ ; The threshold of unfolded distance  $D_\theta$ ; The thresholds of density radio  $\mathfrak{R}_\theta$ ; Splitting criterion; The threshold of the terminal node;
  2. **Output:** An MVU-based CL-DT with multiple condition attributes.
  3. **Process:**
  4. Step 1: The algorithm of MUD: function MUD;
  5. Step 2: The algorithm of the determination of MCL: function MCL;
  6. Step 3: Judge whether co-locations are distinct event types;
  7. Step 4: Build an initial tree;
  8. Step 5: Starting with a single node, the root, which includes all rules and attributes;
  9. Step 6: Judge whether each non-leaf node will be further split, e.g.,  $w_i$ ;
  10.     • Perform a label assignment test to determine if there are any labels that can be assigned;
  11.     • An attribute is selected according to the splitting criterion to split  $w_i$  further and judge whether the stop criterion is met;
  12.         ◦ If the selected attribute meets the splitting criterion, the node will be parted into subsets;
  13.         ◦ If the stop criterion is satisfied, stop splitting and assign  $w_i$  as a leaf node;
  14. Step 7: Apply the MVU-based co-location algorithm for each of the two non-leaf nodes in the same layer, e.g.,  $w_i$  and  $w_j$ , to test whether the two nodes satisfy the co-location criterions. If yes, merge the two neighbor nodes; if no, go to Step 6.
  15. Step 8: Apply the algorithm recursively to each of the not-yet-stopped nodes;
  16. Step 9: Generate decision rules by collecting decisions driven in individual nodes;
  17. Step 10: The decision rules generated in Step 7 are used as the initialization of co-location mining rules. Apply the algorithm of the co-location mining rule to generate new associate rules.
  18. Step 11: Re-organize the input data set and repeat Step 2 through
  19. Step 8 until the classified results by the colocation mining rule and decision tree (rules) are consistent.
- 

## 3. Experiments and Analysis

In this section, we describe the test dataset and experimental environment and present the results of remotely sensed imagery classification. In addition to the proposed method, SAE (a deep neural network), CL-DT, CART, and random forests (RFs) are also compared to evaluate our method. The details are described as follows.

### 3.1. Datasets

(1) The first test area and dataset: The first test area is located from  $24.25^\circ$  to  $26.38^\circ$  north latitude and  $109.60^\circ$  thru  $111.48^\circ$  east longitude, covering the entire city of Guilin, Guangxi Province, China. The area covers approximately  $27,200 \text{ km}^2$ . The test area is a typical karst plain landform in which

there are many exposed carbonate rocks. The images of Landsat-5 TM were acquired at local time on 21 September 2006. The spatial resolution of three visible bands, one near infrared band, one middle infrared band, and one far infrared are 30 m, and the spatial resolution of one thermal infrared band is 120 m. The Landsat-5, with the main detector TM, is operated at a sun-synchronous, 705-km orbit height with a 16-day revisit cycle.

(2) The second test area and dataset: The second test area is located from 23.78° to 24.58° north latitude and 107.85° thru 108.5° east longitude, covering Du'an County of the City of Hechi, Guangxi Province, China. The area covers nearly 4459 km<sup>2</sup>. The test area is a typical karst rocky desertification area in which exposed carbonate rocks are widespread. The images of Landsat-5 TM were acquired at local time on 30 January 2009, and were ordered from the website <http://www.gscloud.cn/>.

(3) The third test area and dataset: The third test area is located from 29.718° to 29.724° north latitude and from 95.346° to 95.354° west longitude. The data were acquired on 23 June 2012 by the NSF-funded Center for Airborne Laser Mapping (NCALM) with a compact airborne spectrographic imager (CASI) over the University of Houston campus. A 311 × 278-pixel subset was used for classification in this paper. The data consists of 144 spectral bands in the 380-nm to 1050-nm region with 2.5-m spatial resolution and had been calibrated to at-sensor spectral radiance units.

These imagery data were pre-processed, including geometric correction [37,38], mosaic [39,40], and clipping [41].

### 3.2. Non-Spatial Attribute and Spatial Attribute Selection

In addition to the original TM imagery data, as a rule of thumb, the data of five non-spatial attributes, including SSM, LST, VC, the components of PCA, and texture (TEX), are considered to classify the land covers. SSM data can be generated by the method of [41]. LST is retrieved from Landsat TM data by a mono-window algorithm [42]. VC can be obtained by the vegetation index method [43]. By applying the PCA function of ENVI 4.8 software for the pre-processed TM imagery data, the components of PCA can be acquired, whose *i*-th component can be represented by PCA<sub>*i*</sub>. Based on the co-occurrence measures, the texture (noted as TEX) data can be produced. Finally, there are 30,217,776 instances for each attribute in the first test area and 4,954,112 instances in the second test area.

For further processing, the instances are considered as five classes, i.e. water (WT), vegetation (VG), exposed carbonatite (EC), habitation (HB), and cultivated land (CL), in the first two experiments.

In addition to the non-geospatial attributes above, spatial attributes, including X and Y coordinates, are considered as well. The metadata for the Geospatial data are listed in Table 1.

**Table 1.** X/Y coordinates.

	Description
X/Y coordinates	Projection: Transverse_Mercator
	False_Easting: 500000.000000
	Central_Meridian: 111.000000
	Scale_Factor: 0.999600
	Latitude_Of_Origin: 0.000000
	Linear Unit: Meter (1.000000)
	Geographic Coordinate System: GCS_WGS_1984
	Angular Unit: Degree (0.017453292519943295)
	Prime Meridian: Greenwich (0.00000000000000)
	Datum: D_WGS_1984

Furthermore, these TM imagery data (SSM, LST, VC, the components of PCA, TEX, and X/Y coordinates) are managed by a database. Each of these coordinates is regarded as one of the dimensions of high-dimensional space in which instances belong to a non-linear distribution. Therefore, the coordinate of every instance can be viewed as a vector expressed by ( $att_{11}, att_{12}, \dots, att_{1n}$ ), where  $att_{ij}$  represents the value of the *i*-th instance of the *j*-th attribute.

### 3.3. Experiments

The flowchart of the experimental procedure is depicted in Figure 5. First, the MVU algorithm is utilized to “unfold” input data, and the MVU-unfolded distances between instances are calculated. Second, the unfolded distance is merged with the co-location mining algorithm to establish the exact RRS between instances, and then the MVU-based co-location rules are mined. Finally, the MVU-based co-location rules are used to induce the generation of the decision tree, and MVU-based co-location decision rules are obtained.

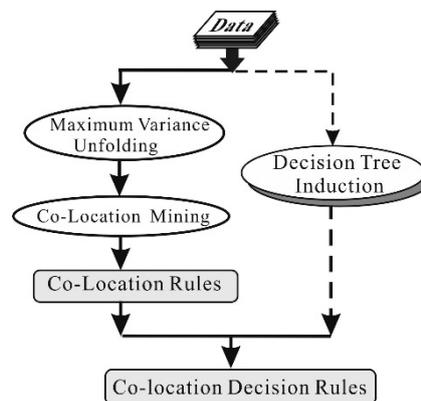


Figure 5. The flowchart of the experiment.

The experiment was conducted using a computer with an Intel Xeon E5645 six-core 2400-MHz CPU (with a 12-MB cache) and 4 GB of RAM. We implemented the proposed algorithm in ENVI + IDL 4.8. For all experiments, we have used 10-fold cross-validation. By connecting with Google Earth, we used polygons, which covered 1 to 4 sample points of the same class in a polygon, to manually collect the samples of five classes (WT, VG, EC, HB, and CL) in Landsat TM images. The number of collected samples containing training samples and testing samples are 10,000 for each class in the 1st and 2nd areas. The Global Moran’s I method [44–47] is used to analyze the spatial autocorrelation of training samples (see Table 2). As shown in Table 2, the training samples are not spatially auto-correlated, which does not affect the classification results. Ten-fold cross-validation breaks datasets of the collected samples into 10 subsets of size  $N/10$ . Nine subsets of the samples of five classes are used to train the proposed classifier, and the one remaining subset is used to test/validate the proposed classifier.

Table 2. The global Moran’s I of training samples under the neighborhood search distance of 20,000 m by ArcMap.

	VC	PCA1	TEX	PCA3	SSM	PCA2
Moran’s I	0.0145	0.0173	0.0156	−0.0209	0.0222	−0.0216
z-Score	0.5902	0.7552	0.6979	−0.6713	0.8306	−0.6334
p-value	0.5549	0.4500	0.4852	0.5020	0.4062	0.5264

#### 3.3.1. Experiments on the First Test Area

##### Input Parameters

**Data input:** The MVU-based co-location database, which has completed the “pre-processing” using the MVU-based co-location mining method, is loaded. Additionally, a subset of data is applied to build the model, and the rest is utilized to validate the performance of the model.

**Parameters input:** It is necessary to input the parameters below to optimize the process of DT induction. These parameters chosen using cross-validation include the following:

1. Minimum node size: The minimum valid node size can be set in the range of 0–100.
2. Maximum purity: If the purity of a node is 95% or higher, the algorithm stops splitting it. Additionally, if its number of records is 1% or less of the total number of records, the algorithm stops splitting it.
3. Maximum depth: The maximum valid depth is 20–30.

Generation of MVU-based co-location mining rules

- Calculation of unfolded distances

There is a high correlation among different bands of remotely sensed imagery, and most of the data are redundant. The principle component analysis (PCA) is able to make these principal component images unrelated with each other. To save computational time in the MVU-based CL-DT generation, this paper proposes applying the PCA method to delete those instances that are apparently not neighbors after eliminating the correlation between images. This paper takes one attribute, PCA<sub>1</sub>, which represents the first components of PCA in image processing, as an example to explain the initial processing.

Different thresholds are set up for the instances of different classes to determinate the initial candidates of the MVU-based co-location. For example, the thresholds of PCA<sub>1</sub> are set as  $r_{\theta 1}$  and  $r_{\theta 2}$  ( $r_{\theta 1} < r_{\theta 2}$ ) for the exposed carbonatite. If the values of PCA<sub>1</sub> are within  $r_{\theta 1}$  and  $r_{\theta 2}$ , the instances with eligible values are the initial candidate instances. This can be mathematically expressed by:

$$PCA_1(i, j) = \begin{cases} PCA_1(i, j) & \text{if } r_{\theta 1} \leq PCA_1(i, j) \leq r_{\theta 2} \\ 0 & \text{other} \end{cases} \quad PCA_1(i, j) \subseteq P \quad (12)$$

where  $r_{\theta 1}$  and  $r_{\theta 2}$  are the given minimum and maximum thresholds of PCA<sub>1</sub>, respectively;  $P$  is the set of PCA<sub>1</sub>( $i, j$ ); and  $i$  and  $j$  represent the  $i$ -th row and  $j$ -th column, respectively.

With the implementation of this step, those non-neighborred instances can be deleted, and the other instances are called “initial candidate instances”. This means that the non-neighborred instances are excluded in the following computation. As a result, computational time is significantly conserved.

From Equation (12), the components of PCA are also selected to determine the initial candidates of the MVU-based co-location. For example, the threshold of PCA<sub>3</sub>, as for the exposed carbonatite, is less than or equal to  $-5$ , i.e.,

$$PCA_3(i, j) = \begin{cases} PCA_3(i, j) & \text{if } PCA_3(i, j) \leq -5 \\ 0 & \text{if } PCA_3(i, j) = \text{other} \end{cases} \quad PCA_3(i, j) \subseteq P \quad (13)$$

When the above step is successfully finished, the MVU algorithm is utilized to unfold these data using Equation (2). After unfolding these data, the unfolded distances between instances are calculated using Equation (7). After calculation, a sparse matrix  $D_{m \times m}$  of unfolded distances among instances can be obtained.

$$D_{m \times m} = \left\{ \begin{array}{ccccc} 0 & 30 & 60 & \dots & 230025 \\ & 0 & 30 & \dots & \vdots \\ & & \dots & \dots & \dots \\ & & & \dots & 30 \\ & & & & 0 \end{array} \right\}_{m \times m} \quad (14)$$

where  $m$  is 30,217,776 and the units are meters.

- Determination of the MVU-based co-location

After obtaining the unfolded distances between instances, the RRSs between instances are determined using Equation (6), in which the threshold of unfolded distance is set as 60, i.e.,  $D_{\theta} = 60$  (m),

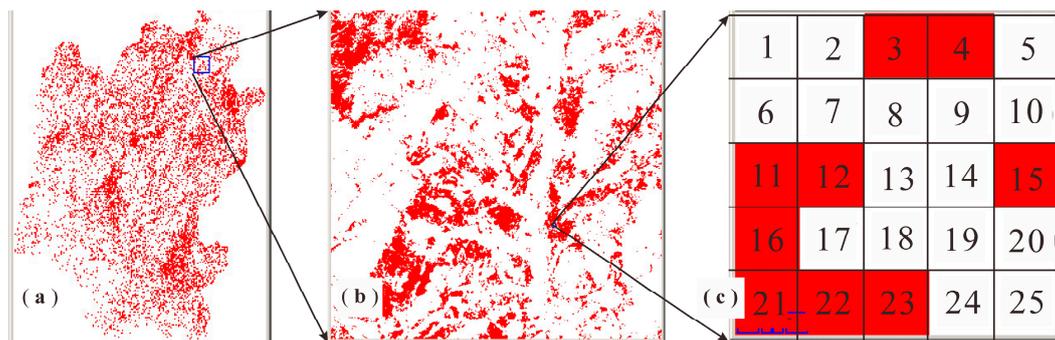
because the resolution of the TM images is 30 m, except for the thermal infrared band. Therefore, we have

$$R(Y_i, Y_j) = \begin{cases} 1 & \text{if } D_U(Y_i, Y_j) \leq 60 \\ 0 & \text{if } D_U(Y_i, Y_j) > 60 \end{cases} \quad (15)$$

With the above determination of the RRS, those instances with RRS are the candidates for the MVU-based co-location. A sparse matrix  $CD_{m \times m}$  of the candidates for MVU-based co-location is generated.

$$CD_{m \times m} = \begin{pmatrix} 0 & 1 & 1 & \cdots & 0 \\ & 0 & 1 & \cdots & \vdots \\ & & \cdots & \cdots & \cdots \\ & & & \cdots & 1 \\ & & & & 0 \end{pmatrix}_{m \times m} \quad (16)$$

On the basis of Equation (7) and Equation (16), the PCA components, VC, SSM, LST, and TEX, of each candidate pair of MVU-based co-location instances are employed to determine the MVU-based co-locations. We take the EC as an example to describe the method here. As mentioned above, if the DRs of candidates of MVU-based co-location of the exposed carbonatite are greater than or equal to 4/5, they are MVU-based co-locations. If not, they will be deleted from the candidates. With the application of this algorithm, the  $k$ th-order MVU-based co-locations can be obtained, as shown in Figure 6.



**Figure 6.** (a) The results of the determination of table instances; (b) the results are amplified two times; and (c) the results are amplified 40 times.

With the calculation above, 8,122,155 instances are preserved. Figure 6c only shows 25 instances, and the 1st-order (ignored) through 6th-order table instances of MVU-based co-location are presented. Table 3 uses four examples to explain the generation processes of the determination of MVU-based co-location. Figure 6c only shows the DRs of the candidate patterns (3, 4) and (11, 12, 16), which are greater than 4/5. For pattern (2, 3), although there is an RRS between them, the TEX is greater than 50, the SSM is not in the range of 5 to 8, and the VC is not less than 0.25 (and similarly so for pattern (10, 15)), so they are deleted from the candidate set.

**Table 3.** Determination of table instances.

Pattern	RRS	PCA3	VC	LST	SSM (%)	TEX	DR
(2, 3)	Yes	(−20.4, −22.3)	(0.33, 0.15)	(17.2, 16.8)	(4.8, 6.8)	(53.7, 46.3)	2/5
(3, 4)	Yes	(−22.3, −18.9)	(0.15, 0.13)	(16.8, 16.3)	(6.8, 7.6)	(46.3, 45.7)	1
(10, 15)	Yes	(−17.2, −20.8)	(0.21, 0.24)	(16.9, 16.8)	(9.8, 5.3)	(58.7, 46.5)	3/5
(11, 12, 16)	Yes	(−19.3, −19.8, −21.2)	(0.21, 0.19, 0.24)	(16.5, 17.1, 16.6)	(6.1, 7.4, 6.7)	(47.6, 46.6, 6.9)	1

- Determination of distinct-type events

On the basis of Equation (8), for EC, the cluster center value of attributes (such as VC) can be obtained from the average value of samples. Thus, we have

$$\Psi_i = (\|f_i - 0.37\|)^2 \quad (17)$$

With the calculation, the threshold is set to 0.04. If  $\Psi_i$  is greater than or equal to 0.04, then the  $i$ -th instance is a distinct event.

### Experimental Results

A DT is induced by the method proposed above. With the post-processing of the DT, five decision rules are induced (Figure 7).

**Rule 1:**  
If (“vegetation coverage”>0.66 and “PCA2>5”) Then “Vegetation”;

**Rule 2:**  
If (“PCA1”>30 and “soil moisture content”>13) Then “Water”

**Rule 3:**  
If (“PCA1”<10 and “PCA3”>10 and “vegetation coverage”<0.4 and “texture”>106)  
Then “Habitation”

**Rule 4:**  
If (“vegetation coverage”>0.4 and “texture”>95 and “PCA1”<20) Then  
“Cultivated land”

**Rule 5:**  
If (“PCA3”<5 and “soil moisture content”>5 and “texture”<94 and “land surface temperature”>16)  
Then “Bared carbonate rock”

**Figure 7.** The final rules after verification and post-processing.

The feature/attribute importance can be measured by the importance rate of attribute (IRA) [48]. Suppose  $n$  attributes,  $(att_1, att_2, \dots, att_n)$ , are used to build the DT. The total number,  $num\_att_i$ , of training samples classified using the  $i$ -th attribute  $att_i$  can be obtained by:

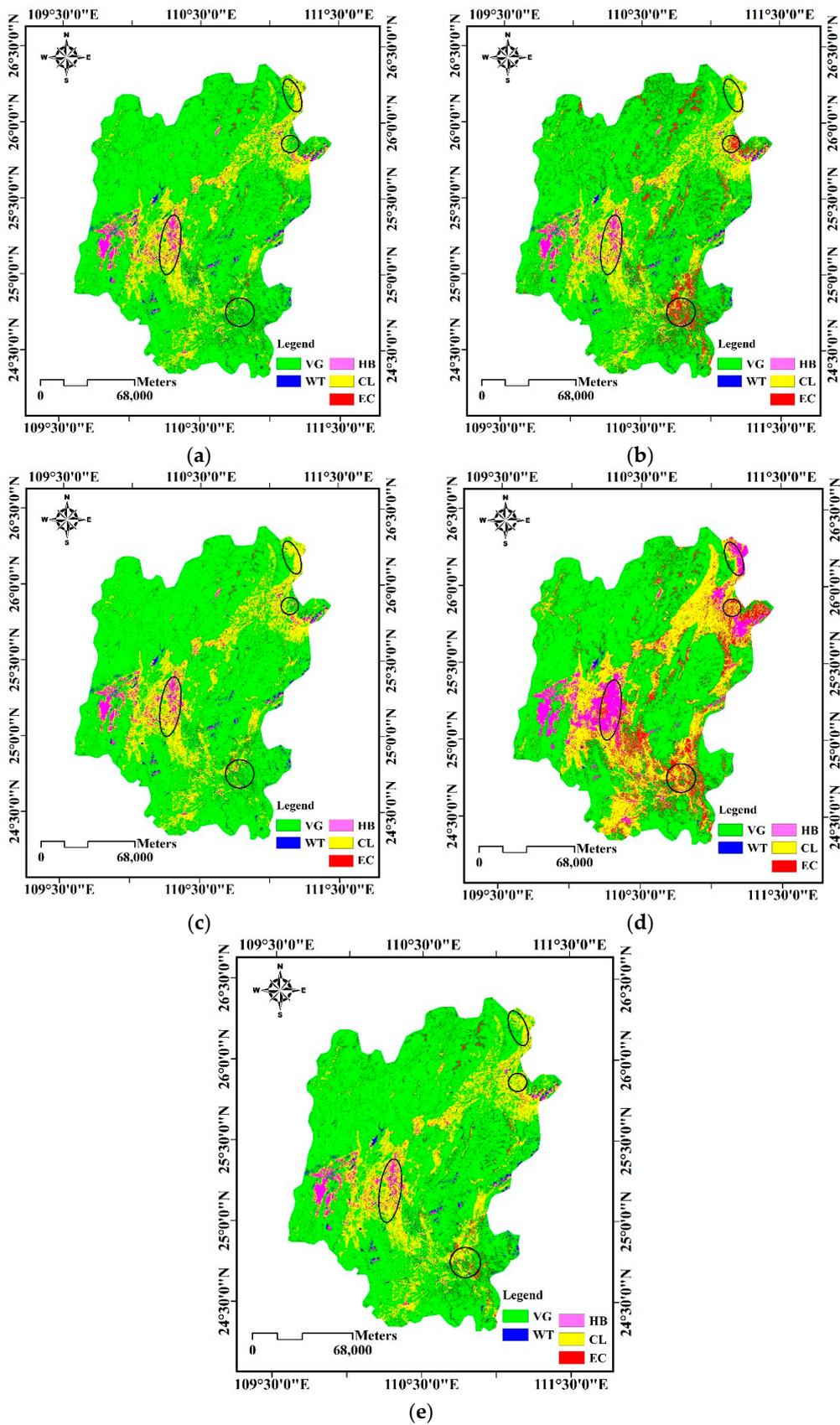
$$num\_att_i = \sum_j^m splitnum(j) \quad (18)$$

where  $m$  is the time of classification participation of the  $i$ -th attribute and  $splitnum(j)$  is the number of training samples classified using  $att_i$  at the  $j$ -th time. Thus, the importance rate of the  $i$ -th attribute,  $IRA_i$ , can be calculated by:

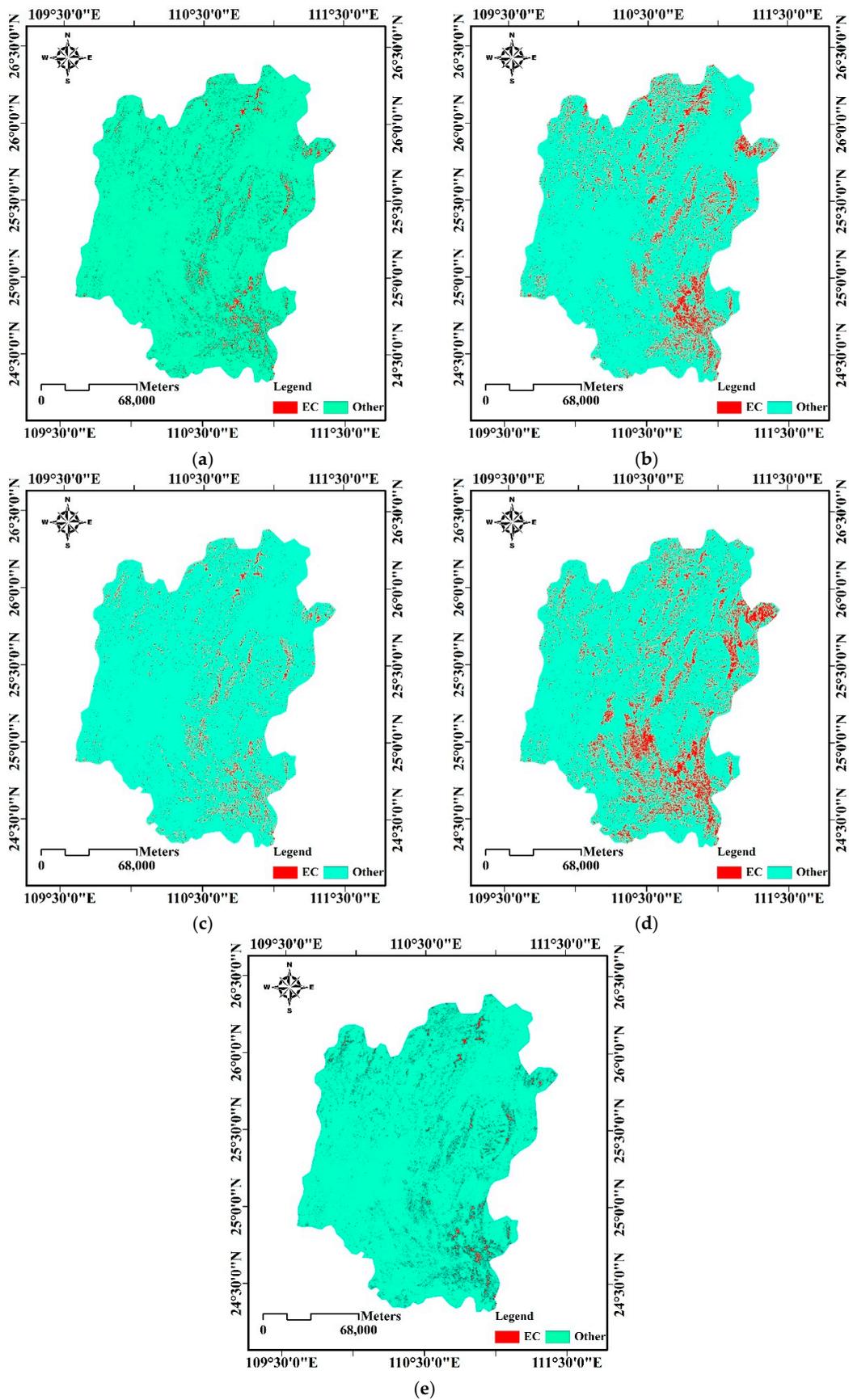
$$IRA_i = \frac{num\_att_i}{\sum_i num\_att_i} \quad (19)$$

After calculation, the importance rates of attributes' classification participation are 0.375, 0.208, 0.129, 0.117, 0.083, 0.051, and 0.037 for VC, PCA1, TEX, PCA3, SSM, LST, and PCA2, respectively.

The classification results in the first test area using the proposed method are presented in Figure 8a. For comparison analysis, the classified results using CL-DT, SAE [1,49], CART and RFs [50] methods are depicted in Figure 8b–e, respectively. Furthermore, their differences are marked by black circles in Figure 8. In addition, the distribution of exposed carbonatite is shown in Figure 9.



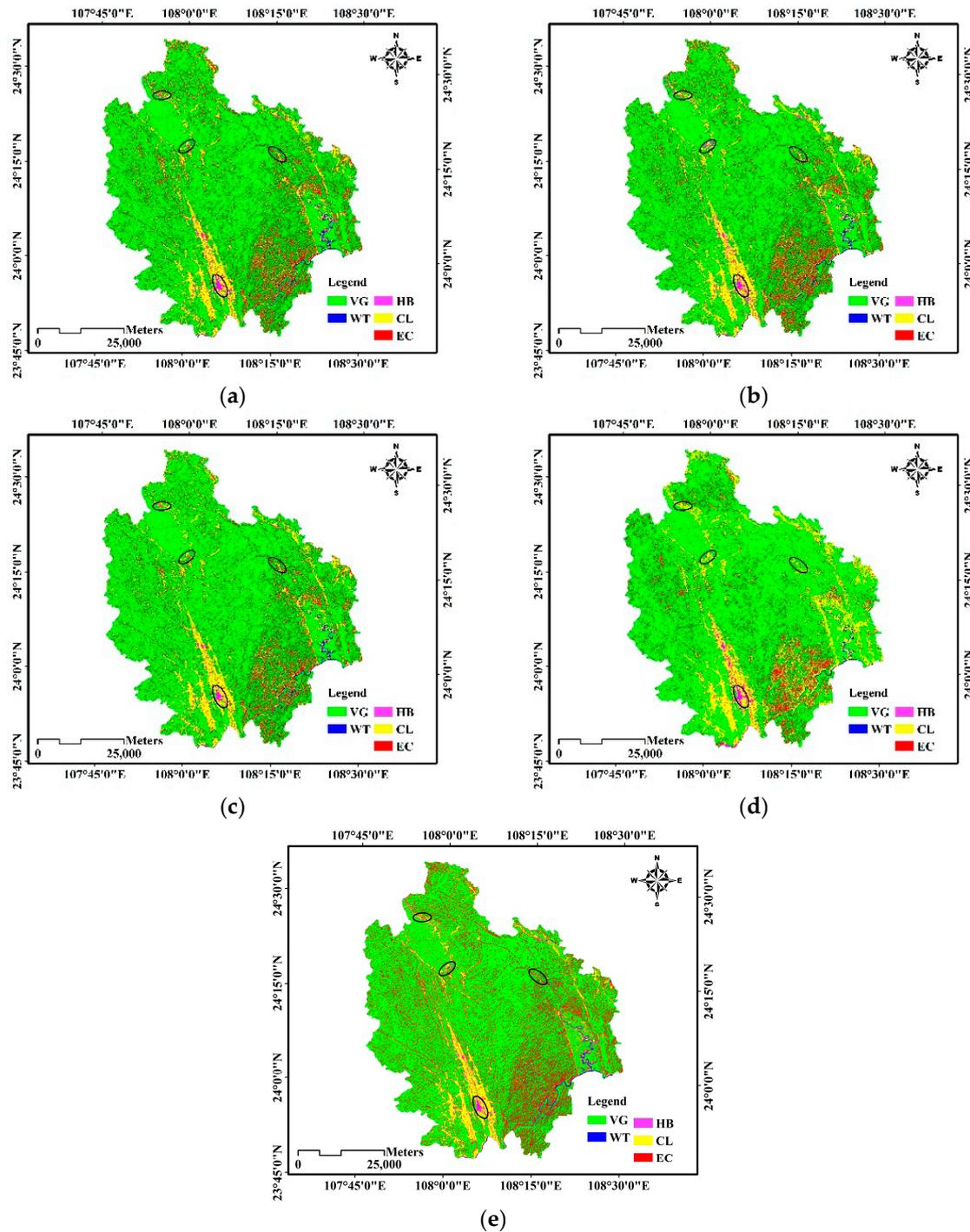
**Figure 8.** The results of classification in the first test area by: (a) the proposed method; (b) co-location decision tree (CL-DT); (c) stacked auto-encoders (SAE); (d) classification and regression tree (CART); and (e) random forests (RFs).



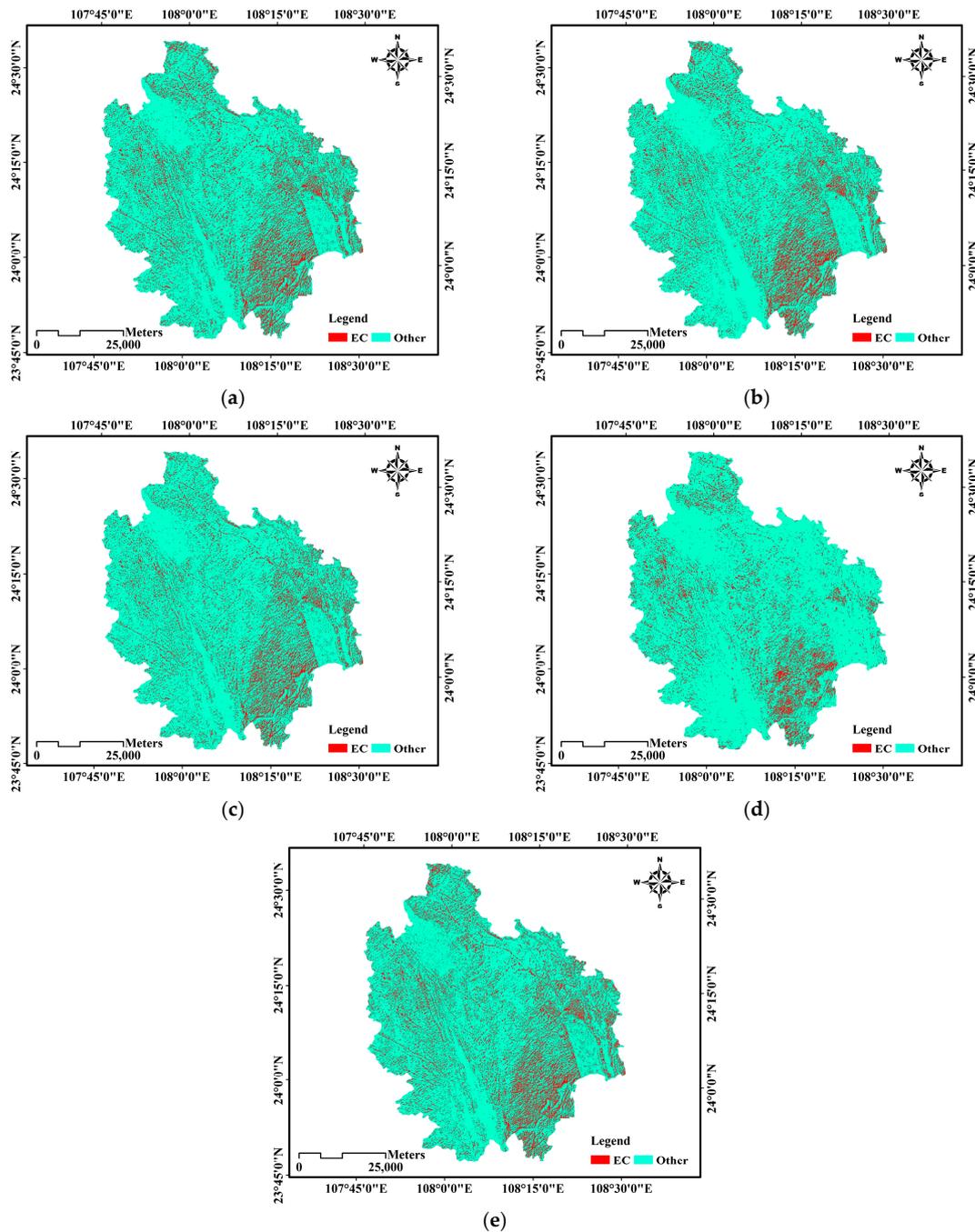
**Figure 9.** The extraction results of exposed carbonatite in the first test area by: (a) the proposed method; (b) CL-DT; (c) SAE; (d) CART; and (e) RFs.

### 3.3.2. Experiments on the Second Test Area

Similarly, the classification results in the second test area using the proposed method and the CL-DT, SAE, DT, and RFs methods are shown in Figure 10a–e, respectively. Furthermore, their differences are marked with black circles in Figure 10. In addition, the distribution of exposed carbonatite in the second test area is shown in Figure 11.



**Figure 10.** The results of classification in the second test area by: (a) the proposed method; (b) CL-DT; (c) SAE; (d) CART; and (e) RFs.



**Figure 11.** The extraction results of exposed carbonatite in the second test area by: (a) the proposed method; (b) CL-DT; (c) SAE; (d) CART; and (e) RFs.

### 3.3.3. Experiments on the Third Test Area

To obtain a similar amount of bands with Landsat-5, the blue channel was simulated by the 21st–34th bands of the hyperspectral dataset, the green channel was simulated by the 34th–42nd bands, the red channel was simulated by the 57th–70th bands, and the near-infrared channel was simulated by the 84th–113th bands (see Figure 12). Through the methods mentioned in Section 3.2, VC, TEX, and PCA can be obtained. The spatial attribute is also described in X and Y coordinates. According to the training samples given by NCALM, the instances are considered as the following five classes: vegetation (VG), building (BD), soil (SD), parking lot (PL), and road (RD). The number of samples, containing training and validation samples, is 300 for each class.

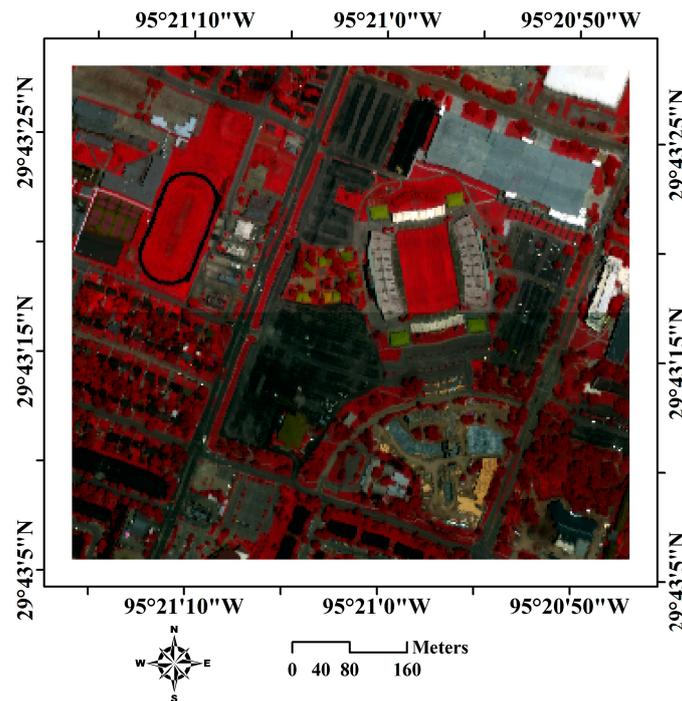


Figure 12. False color image of the 3rd test area.

Similarly, the threshold of distance  $D_\theta$  and density ratio  $RD_\theta$  are 5 m and  $6/7$ , respectively. The final rules are shown in Figure 13. The classification results in the third test area using the MVU-based CL-DT, CL-DT, SAE, DT, and RFs methods are shown in Figure 14a–e, respectively.

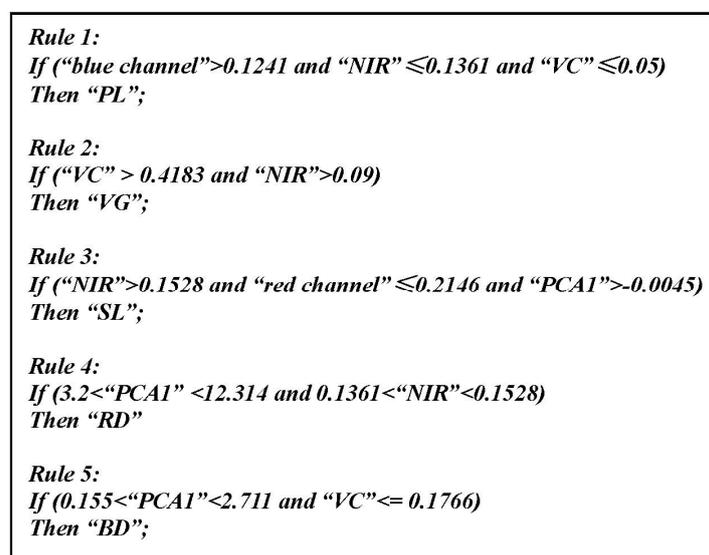


Figure 13. The final rules of the third test area (R: NIR channel; G: red channel; B: green channel).

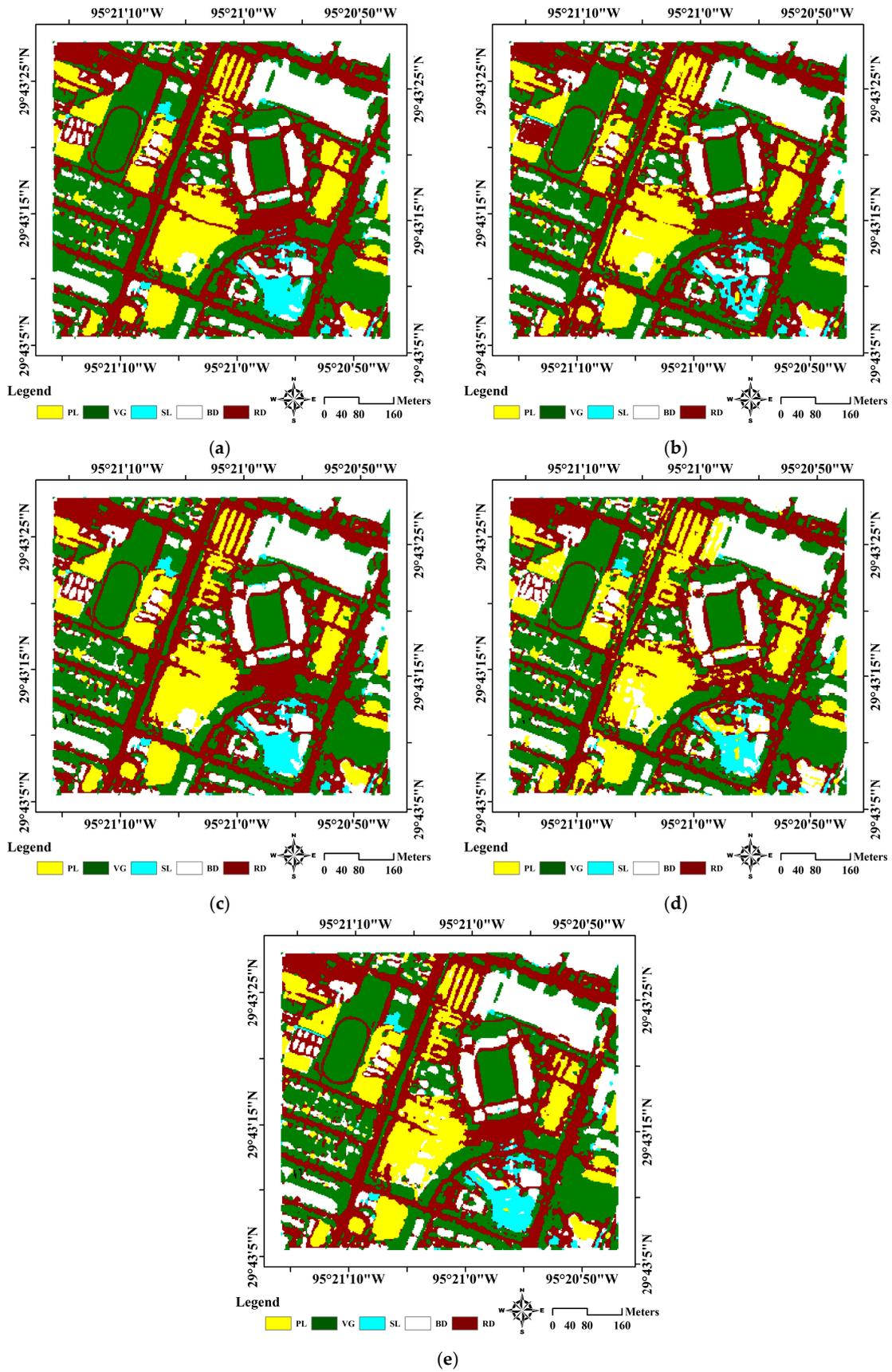


Figure 14. The results of classification in the third test area by: (a) the proposed method; (b) CL-DT; (c) SAE; (d) CART; and (e) RFs.

## 4. Comparison Analysis and Validation in the Field

### 4.1. Classification Accuracy Comparison

To analyze the classification accuracy of the proposed MVU-based CL-DT method compared to other methods, such as SAE, CL-DT, CART, and RFs, this paper takes the classification results from the SAE method as “true values” and conducts statistical analysis using ENVI 4.8 software. Compared to the “true values,” the relative difference (RD) of classification results and relative accuracy (RA) for test areas 1, 2, and 3 are shown in Tables 4–6, respectively.

**Table 4.** The relative difference (RD) of classification results and relative accuracy (RA) in the 1st test area.

	SAE (Baseline)	DT	CL-DT	RFs	MVU-Based CL-DT
RD (%)	0	25.32	16.38	9.43	8.93
RA (%)	100	74.68	83.62	90.57	91.07

**Table 5.** The RD of classification results and RA in the 2nd test area.

	SAE (Baseline)	DT	CL-DT	RFs	MVU-Based CL-DT
RD (%)	0	25.03	14.72	9.08	8.46
RA (%)	100	74.97	85.28	90.92	91.54

**Table 6.** The RD of classification results and RA in the 3rd test area.

	SAE (Baseline)	DT	CL-DT	RFs	MVU-Based CL-DT
RD (%)	0	15.89	7.77	4.62	3.61
RA (%)	100	84.11	92.23	95.38	96.39

To further analyze the classification accuracy, the proportions of the five classes are statistically analyzed and shown in Figure 15. As observed from Figure 15a, the proportions of each class classified by SAE and MVU-based CL-DT are very close. For example, their smallest difference is approximately 0.2% for WT, and their largest difference is 1.1% for EC in the first test area.

As shown in Tables 4–6 and Figure 15, the proposed MVU-based CL-DT has the highest classification accuracy relative to the “true value” and that the CART has the worst accuracy.

To further explain the impact of the geospatial relationship for classification accuracy, spectral curves and magnified windows of classification results of MVU-based CL-DT, CL-DT, CART, and RFs are employed for visual verification (see Figures 16 and 17). As observed from Figure 16, although the walnut, pine, and grass families are three different features, their spectral curves are very similar. One of them, pine (masson pine), is the main natural vegetation in the first test area, which is represented as “VG” in the classification results. The grass family (for example, rice) and walnut are the main crops, which are represented as “CL” in classification results. Because their spectral characteristics are so close, the CART method misclassifies “VG” and “CL.” For example, as observed from Figure 17, the classification results of SAE and MVU-based CL-DT are the closest to the ground truth. The classification results of the CART method are significantly different from the reality in the remotely sensed imagery.

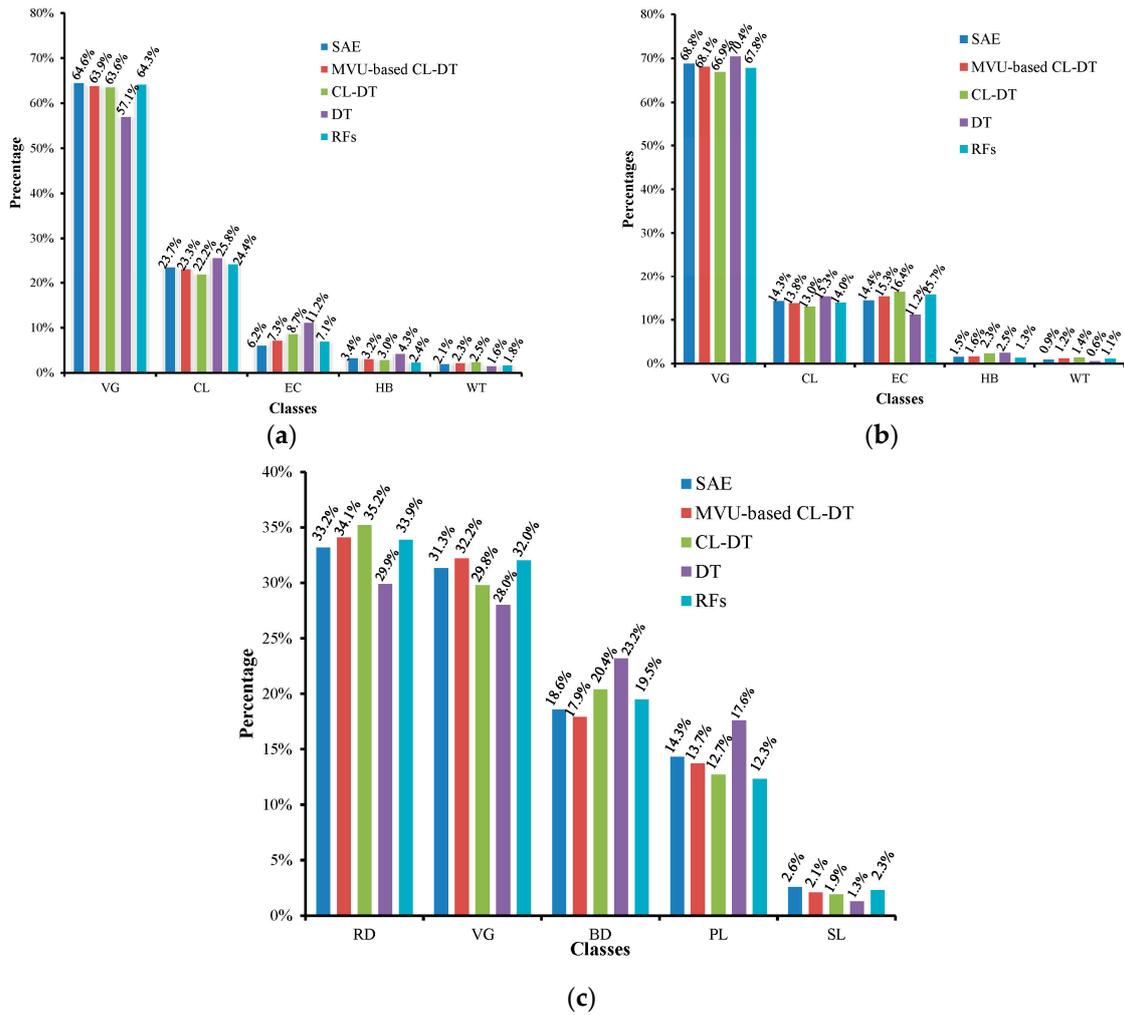


Figure 15. Comparison of the proportion of classes retrieved by different methods in: the first test area (a); second test area (b); and third test area (c).

The experimental results above have demonstrated that the proposed MVU-based CL-DT method is capable of effectively decreasing misclassification, which is especially caused by images with different objects but the same spectra, and has a higher classification accuracy than CART and CL-DT.

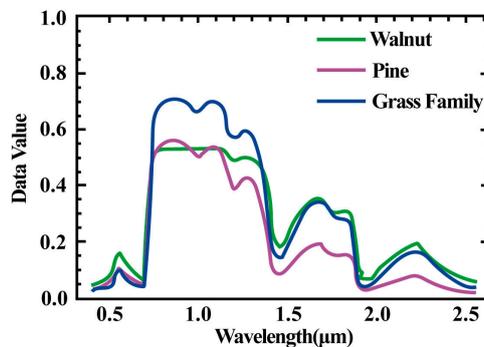
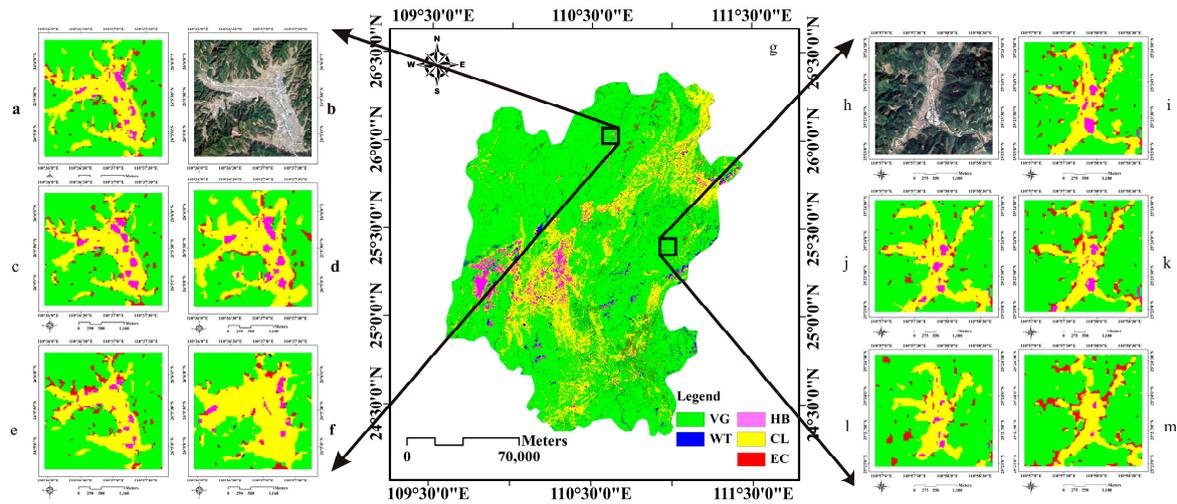


Figure 16. Spectral curves of features (note: these spectral curves of features are from the spectral libraries of ENVI 4.8).



**Figure 17.** (a–m) Magnified windows of classification results in the first test area: (g) the classification results of SAE; (b,h) ground truth; and magnified windows of the classification results of: (a,i) MVU-based CL-DT; (d,j) CL-DT; (c,k) SAE; (f,l) CART; and (e,m) RFs.

In addition, to further demonstrate the advantages and accuracy improvement in classification for the proposed method, independent ground truth samples from Google Earth and confusion matrixes (Tables 7–21) are employed to create the producer accuracy (PA), user accuracy (UA), over-accuracy (OA), average accuracy (AA), and Kappa coefficient (KC). The comparison analyses of five types of methods are presented in Table 22.

**Table 7.** Confusion matrix of the proposed method in the first area.

	VG	WT	EC	HB	CL	PA	UA
VG	967	49	38	0	65	0.967	0.864
WT	0	945	0	0	0	0.945	1.00
EC	29	2	839	42	74	0.839	0.850
HB	0	0	0	958	0	0.958	1.00
CL	4	4	123	0	861	0.861	0.868
AA						0.914	0.916

**Table 8.** Confusion matrix of the proposed method in the second area.

	VG	WT	EC	HB	CL	PA	UA
VG	918	7	29	0	77	0.918	0.890
WT	0	967	0	0	0	0.967	1.00
EC	65	0	809	65	32	0.809	0.833
HB	8	6	25	929	8	0.929	0.952
CL	9	20	137	6	883	0.883	0.837
AA						0.901	0.902

**Table 9.** Confusion matrix of the proposed method in the third area.

	PL	VG	SL	BD	RD	PA	UA
PL	25	0	0	2	3	0.833	0.833
VG	0	30	0	0	0	1.00	1.00
SL	1	0	23	0	1	0.767	0.920
BD	2	0	4	27	0	0.900	0.818
RD	2	0	3	1	26	0.867	0.837
AA						0.873	0.882

**Table 10.** Confusion matrix of the CL-DT in the first area.

	VG	WT	EC	HB	CL	PA	UA
VG	864	46	61	0	29	0.864	0.864
WT	9	893	0	56	13	0.893	0.920
EC	37	9	793	169	119	0.793	0.704
HB	23	31	4	753	0	0.753	0.928
CL	67	21	142	22	839	0.839	0.769
AA						0.828	0.837

**Table 11.** Confusion matrix of the CL-DT in the second area.

	VG	WT	EC	HB	CL	PA	UA
VG	864	53	137	48	62	0.864	0.742
WT	12	909	0	24	73	0.909	0.893
EC	28	7	753	27	107	0.753	0.817
HB	19	2	67	819	8	0.819	0.895
CL	77	29	43	82	750	0.750	0.765
AA						0.819	0.822

**Table 12.** Confusion matrix of the CL-DT in the third area.

	PL	VG	SL	BD	RD	PA	UA
PL	22	0	0	2	5	0.733	0.759
VG	0	29	0	1	0	0.967	0.967
SL	2	1	24	0	1	0.800	0.857
BD	2	0	3	25	1	0.833	0.807
RD	4	0	3	2	23	0.767	0.719
AA						0.820	0.822

**Table 13.** Confusion matrix of the CART in the first area.

	VG	WT	EC	HB	CL	PA	UA
VG	1000	41	25	5	29	1.00	0.909
WT	0	728	0	6	4	0.728	0.987
EC	0	115	793	131	241	0.793	0.619
HB	0	109	57	847	92	0.847	0.766
CL	0	7	125	11	634	0.634	0.816
AA						0.800	0.819

**Table 14.** Confusion matrix of the CART in the second area.

	VG	WT	EC	HB	CL	PA	UA
VG	953	4	47	0	0	0.953	0.949
WT	0	954	0	0	0	0.954	1.00
EC	45	31	500	100	18	0.500	0.720
HB	0	0	0	713	0	0.713	1.00
CL	2	11	453	187	982	0.982	0.601
AA						0.820	0.854

**Table 15.** Confusion matrix of the CART in the third area.

	PL	VG	SL	BD	RD	PA	UA
PL	22	0	1	0	8	0.733	0.710
VG	0	27	0	2	0	0.900	0.931
SL	1	2	21	0	0	0.700	0.875
BD	0	1	7	23	1	0.767	0.719
RD	7	0	1	5	21	0.700	0.618
AA						0.760	0.771

**Table 16.** Confusion matrix of the SAE in the first area.

	VG	WT	EC	HB	CL	PA	UA
VG	996	0	0	0	0	0.996	1.00
WT	0	1000	19	0	0	1.00	0.981
EC	4	0	893	53	24	0.893	0.917
HB	0	0	88	947	0	0.947	0.915
CL	0	0	0	0	976	0.976	1.00
AA						0.962	0.963

**Table 17.** Confusion matrix of the SAE in the second area.

	VG	WT	EC	HB	CL	PA	UA
VG	1000	0	15	0	23	1.00	0.963
WT	0	1000	0	0	0	1.00	1.00
EC	0	0	954	14	120	0.954	0.877
HB	0	0	10	986	4	0.986	0.986
CL	0	0	21	0	853	0.853	0.976
AA						0.959	0.960

**Table 18.** Confusion matrix of the SAE in the third area.

	PL	VG	SL	BD	RD	PA	UA
PL	26	0	0	1	2	0.866	0.897
VG	0	30	0	0	0	1.00	1.00
SL	1	0	25	0	1	0.833	0.926
BD	2	0	2	28	0	0.933	0.875
RD	1	0	3	1	27	0.900	0.844
AA						0.906	0.908

**Table 19.** Confusion matrix of the RFs in the first area.

	VG	WT	EC	HB	CL	PA	UA
VG	949	0	19	0	31	0.949	0.950
WT	2	964	0	6	7	0.964	0.985
EC	14	0	819	63	116	0.819	0.809
HB	0	27	49	931	0	0.931	0.925
CL	35	9	113	0	846	0.846	0.843
AA						0.902	0.902

**Table 20.** Confusion matrix of the RFs in the second area.

	VG	WT	EC	HB	CL	PA	UA
VG	932	13	24	0	17	0.932	0.945
WT	0	944	0	19	0	0.944	0.980
EC	24	0	838	35	112	0.838	0.831
HB	11	31	17	937	10	0.937	0.931
CL	33	12	121	9	861	0.861	0.832
AA						0.902	0.904

**Table 21.** Confusion matrix of the RFs in the third area.

	PL	VG	SL	BD	RD	PA	UA
PL	24	0	2	1	3	0.80	0.80
VG	0	29	0	0	0	0.967	1.00
SL	1	1	24	0	2	0.80	0.857
BD	3	0	2	28	0	0.933	0.848
RD	2	0	2	1	25	0.833	0.833
AA						0.867	0.868

**Table 22.** Comparison of over accuracy (OA) and Kappa coefficient (KC) in the three test areas.

	1st Test Area		2nd Test Area		3rd Test Area	
	OA	KC	OA	KC	OA	KC
Our method	91.40%	0.89	90.12%	0.87	87.34%	0.84
CL-DT	82.84%	0.79	81.80%	0.77	82.0%	0.78
SAE	96.24%	0.95	95.86%	0.94	90.66%	0.88
CRAT	80.04%	0.75	82.04%	0.78	76.0%	0.70
RFs	90.18%	0.88	90.24%	0.88	86.67%	0.83

As mention in Section 1, the major shortcoming of the CL-DT method is that it only considers the Euclidean distance of neighborhood instances, which cannot truly reflect the real co-location relationship between instances for which there is a nonlinear distribution in a high-dimensional space. It has been demonstrated that MVU can preserve the original neighborhood relationship of instances which are with nonlinear distribution in a high-dimensional space [30]. Obviously, the proposed MVU-based CL-DT method takes advantage of the characteristics of MVU to obtain the unfolded distance of instances. Therefore, the proposed method is capable of in fact ensuring that the co-location relationship of instances is real. As a consequence, the proposed method overcomes the shortcoming of CL-DT method, and, meanwhile, inherits all advantages of the CL-DT method. After using accuracy, which is assessed by testing the induced DT with new data sets and then comparing the predicted classes with the real classes, to evaluate the quality of DT, it is shown that the proposed MVU-based CL-DT can create a higher accuracy of DT than both the CL-DT does and traditional DT does because it uses unfolded distances to determine the co-locations.

#### 4.2. Comparison Analysis for Parameters and Time-Consumption

To further evaluate the advantages of the proposed method, the induced DT parameters and computational time are compared to those from MVU-based CL-DT, CL-DT, CART, RFs, and SAE below.

- Comparison of the induced decision tree parameters

The DT parameters induced by the proposed method, CL-DT method, CART, and RFs are compared, and the results are listed in Table 23. As shown in Table 23, the total number of nodes, number of leaf nodes, and number of levels from the proposed method, when compared to CL-DT, decrease by 48%, 45%, and 25%, respectively. The total number of nodes, number of leaf nodes,

and number of levels of the proposed method, when compared to CART, decrease by 56%, 54%, and 33%, respectively. The number of trees to RFs is 100, while to MVU-base CL-DT is 1. This means that the proposed MVU-based CL-DT algorithm is able to construct a simpler DT than the others.

**Table 23.** Comparison of the induced decision tree parameters.

	DT	CL-DT	Our Method
Total number of nodes	25	21	11
Number of leaf nodes	13	11	6
Number of levels	9	8	6

- Comparison of the computational time

The running time of the proposed method consists of two parts, i.e., the pre-processing time (unfolding data sets) and the time of the DT construction. In the pre-processing phase, several minutes are taken to unfold the original dataset (i.e., large images). However, in the DT construction phase, the time can be largely decreased, which has been reported in Table 24. As observed in Table 24, the time taken for rule generation decreases by 43% and 59%, respectively, when compared to CL-DT and the CART. Moreover, the total running time of SAE is 113 s. Additionally, RFs needs 33 s to do parameterization and more than 300 s to classify images. This result demonstrates that the proposed method has faster computational speed in constructing decision trees than the other methods.

Through the comparison of complexity indicated to the shape and size of a DT, it is demonstrated that the proposed MVU-based CL-DT can create a simpler and better DT than both the CL-DT does and traditional DT does.

**Table 24.** Comparison of the computational time of constructing the DT.

	Time Taken (Seconds)		
	DT	CL-DT	Our Method
Data processing	6	4	3
Decision tree growing	11	8	5
Decision tree drawing	22	18	10
Generating rules	52	37	21

#### 4.3. Classification Accuracy Validation in the Field

For the first test area, we conducted a field validation of the classification accuracy in Guanyang County, Guilin, China. We use a Magellan 210 GPS to collect 20 field validation samples of exposed carbonatite, including their longitudes and latitudes (Table 25).

**Table 25.** Sample points.

	Latitude	Longitude		Latitude	Longitude
1	25°36'11.52"	111°6'38.88"	11	25°36'39.45"	111°6'49.18"
2	25°36'20.29"	111°6'47.49"	12	25°36'38.63"	111°6'53.45"
3	25°36'17.86"	111°6'39.42"	13	25°36'32.58"	111°6'47.21"
4	25°36'25.42"	111°6'38.86"	14	25°36'39.41"	111°6'57.76"
5	25°36'26.88"	111°6'41.03"	15	25°36'33.75"	111°6'44.85"
6	25°36'25.66"	111°6'1.01"	16	25°36'37.07"	111°6'51.73"
7	25°36'33.95"	111°6'48.99"	17	25°36'40.58"	111°6'32.31"
8	25°36'35.41"	111°6'51.41"	18	25°36'41.16"	111°6'36.61"
9	25°36'37.85"	111°6'48.72"	19	25°36'32.56"	111°6'37.25"
10	25°36'36.39"	111°6'47.11"	20	25°36'25.92"	111°6'45.42"

These field validation samples are used to validate the classification results obtained by the proposed method. Only two samples, samples 1 and 2, do not match the classified results, yielding an error proportion of 10%, i.e., a 90% correctness ratio.

## 5. Conclusions

The primary contribution of this research is the proposal of the maximum variance unfolding (MVU)-based co-location decision tree (MVU-based CL-DT) algorithm. The algorithm overcomes the deficiency of the traditional CL-DT method, in which the Euclidean distance of instances that are nonlinear distributions in high-dimensional space cannot accurately reflect the co-location relationship between instances. With the experimental results and comparison analysis in Section 4, it can be concluded that the proposed method is capable of effectively and successfully overcoming the classification problems, i.e., “the different objects but equivalent/similar spectrum” issue, that the remote sensing community has traditionally encountered.

This paper has provided detailed descriptions and steps for the algorithms. First, the MVU algorithm is utilized to unfold the input data, and the unfolded distances between instances in the unfolded data are calculated. Second, according to the unfolded distances, the *R*-relationship (RRS) between instances is determined. Third, MVU-based co-location instances are found on the basis of the RRS, the distinct events are determined, and the MVU-based co-location rules are generated. Finally, the MVU-based co-location rules are merged into the DT to generate the MVU-based co-location decision rules.

The proposed method has been used to classify remotely sensed imagery in three test areas. When compared to stacked auto-encoders (SAE), whose classification results are taken as “true values,” CL-DT, traditional DT (classification and regression tree, i.e., CART), and random forests (RFs), the results herein have demonstrated that: (1) the proposed method has better classification accuracy, with relative accuracies of 91.07%, 91.54%, and 96.39% in the three test areas (with the CL-DT method achieving 83.62%, 85.28% and 92.23%, the CART yielding only 74.68%, 74.97% and 84.11% and RFs obtaining 90.57%, 90.92% and 95.38%); and (2) the proposed method can produce a better tree because the total number of nodes, the number of leaf nodes, and the number of levels of the proposed method decrease by 48%, 45%, and 25%, respectively, compared to the CL-DT method and decrease by 56%, 54%, and 33% compared to the CART method. The computational time taken for data processing, decision tree construction, and rule generation decreases by 25%, 38%, 44%, and 43%, respectively, compared to CL-DT and decreases by 50%, 55%, 55%, and 60% compared to the CART.

With the calculation results of the confusion matrix using the validation samples, it can be concluded that: (1) the over-accuracies (OAs) for MVU-based CL-DT reach 91.4%, 90.1% and 87.34% and the Kappa coefficients are 0.89, 0.87 and 0.84 in the three test areas; (2) the OAs for SAE achieve 96.24%, 95.86% and 90.66% and the Kappa coefficients are 0.95, 0.94 and 0.88 in the three test areas; (3) the OAs for CL-DT achieve 82.84%, 81.80% and 82.0% and the Kappa coefficients are 0.79, 0.77 and 0.78 in the three test areas; (4) the OAs for DT reach 80.04%, 82.04% and 76.01% and the Kappa coefficients are 0.75, 0.78 and 0.70 in the three test areas; and (5) the OAs for RFs achieve 90.18%, 90.24% and 86.67% and the Kappa coefficients are 0.88, 0.88 and 0.83 in the three test areas.

The proposed method combines the MVU algorithm, co-location mining algorithm, and decision tree algorithm. On the one hand, the proposed method can produce a better tree than traditional DT and obtain satisfactory classification results; on the other hand, it appears that the proposed algorithm is somewhat complicated. People are required to have professional knowledge in the three fields to understand the details of the proposed algorithm.

**Acknowledgments:** The authors would like to thank the Hyperspectral Image Analysis group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the data sets used in this study. This paper is financially supported by the National Natural Science Foundation of China (NSFC) (Grant No. #: 41431179, 41162011), GuangXi Governor (Grant No. #: 2010-169), GuangXi Grand Natural Science Foundation (Grant No. #: 2015GXNSFDA139032 and 2012GXNSFCB053005), Guangxi Science & Technology Development Program (Contract No. #: GuiKeHe 14123001-4), and GuangXi Key Laboratory of

Spatial Information and Geomatics Program (Contract No. #: GuiKeNeng 140452401, 140452409, 151400701 and 151400712), the “Ba Gui Scholars” program of the provincial government of Guangxi, Regional Demonstration Project of Marine Economic Innovation and Development of State Oceanic Administration (Project No. #: cxsf-39).

**Author Contributions:** G.Z. contributes most to this manuscript. He contributed the whole idea of this manuscript and designed the experiments and reviewed and revised this manuscript; R.Z. performed the experiments and analyzed the data; G.Z. and R.Z. wrote the paper; D.Z. contributed to editing and reviewed this manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Chen, Y.S.; Lin, Z.H.; Zhao, X.; Wang, G.; Gu, Y.F. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**. [[CrossRef](#)]
- Huang, X.; Zhang, L.P. An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 257–272. [[CrossRef](#)]
- Huang, X.; Weng, C.L.; Lu, Q.K.; Feng, T.T.; Zhang, L.P. Automatic labelling and selection of training samples for high-resolution remote sensing image classification over urban areas. *Remote Sens.* **2015**, *7*, 16024–16044. [[CrossRef](#)]
- Huang, X.; Lu, Q.K.; Zhang, L.P. A multi-index learning approach for classification of high-resolution remotely sensed images over urban areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *90*, 36–48. [[CrossRef](#)]
- Simard, M.; Saatchi, S.S.; Grandi, G.D. Use of decision tree and multi-scale texture for classification of JERS-1 SAR data over tropical forest. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 2310–2321. [[CrossRef](#)]
- Franklin, S.E.; Stenhouse, G.B.; Hansen, M.J.; Popplewell, C.C.; Dechka, J.A.; Peddle, D.R. An Integrated Decision Tree Approach (IDTA) to mapping land cover using satellite remote sensing in support of grizzly bear habitat analysis in the Alberta yellow head ecosystem. *Can. J. Remote Sens.* **2001**, *27*, 579–592. [[CrossRef](#)]
- Zhou, G.; Wang, L.; Wang, D.; Reichle, S. Integration of GIS and data mining technology to enhance the pavement management decision making. *J. Transp. Eng.* **2010**, *136*, 332–341. [[CrossRef](#)]
- Xu, C.G.; Anwar, A. Based on the decision tree classification of remote sensing image classification method application. *Appl. Mech. Mater.* **2013**. [[CrossRef](#)]
- Chasmer, L.; Hopkinson, C.; Veness, T.; Quinton, W.; Baltzer, J. A decision-tree classification for low-lying complex land cover types within the zone of discontinuous permafrost. *Remote Sens. Environ.* **2014**, *143*, 73–84. [[CrossRef](#)]
- Wu, C.; Landgrebe, D.; Swain, P. *The Decision Tree Approach to Classification*; NASA-CR-141930, LARS-INFORM-NOTE-090174, TR-EE-75-17; Purdue University: West Lafayette, IN, USA, 1 May 1975.
- Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Wadsworth International Group: Belmont, CA, USA, 1984.
- Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1987**. [[CrossRef](#)]
- Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Mateo, CA, USA, 1993.
- Bujlow, T.; Riaz, M.T.; Pedersen, J.M. A method for classification of network traffic based on C5.0 machine learning algorithm. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC) 2012, Maui, HI, USA, 30 January–2 February 2012; pp. 237–241.
- Polat, K.; Gunes, S. A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Syst. Appl.* **2009**, *36*, 1587–1592. [[CrossRef](#)]
- Franco-Arcega, A.; Carrasco-Ochoa, J.A.; Sanchez-Diaz, G.; Martinez-Trinidad, J.F. Decision tree induction using a fast splitting attribute selection for large datasets. *Expert Syst. Appl.* **2011**, *38*, 14290–14300. [[CrossRef](#)]
- Aviad, B.; Roy, G. Classification by clustering decision tree-like classifier based on adjusted clusters. *Expert Syst. Appl.* **2011**. [[CrossRef](#)]
- Sok, H.K.; Ooi, M.P.; Kuang, Y.C. Sparse alternating decision tree. *Pattern Recognit. Lett.* **2015**. [[CrossRef](#)]
- Zhou, G.; Wang, L. Co-location decision tree for enhancing decision—Making of pavement maintenance and rehabilitation. *Transport. Res. Part C Emerg. Technol.* **2012**, *21*, 287–305. [[CrossRef](#)]
- Mansour, Y. Pessimistic decision tree pruning based on tree size. In Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, USA, 8–12 July 1997.

21. Osei-Bryson, K. Post-pruning in decision tree induction using multiple performance measures. *Comput. Oper. Res.* **2007**, *34*, 3331–3345. [[CrossRef](#)]
22. Osei-Bryson, K. Post-pruning in regression tree induction: An integrated approach. *Expert Syst. Appl.* **2008**, *34*, 1481–1490. [[CrossRef](#)]
23. Balamurugan, S.A.; Rajaram, R. Effective solution for unhandled exception in decision tree induction algorithms. *Expert Syst. Appl.* **2009**, *36*, 12113–12119. [[CrossRef](#)]
24. Appel, R.; Fuchs, T.; Dollar, P.; Peronal, P. Quickly boosting decision trees: Pruning underachieving features early. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013.
25. Li, X.; Xing, Q.; Kang, L. Remote sensing image classification method based on evidence theory and decision tree. *Proc. SPIE* **2010**. [[CrossRef](#)]
26. Farid, D.; Zhang, L.; Rahman, C.; Hossain, M.A.; Strachan, R. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Syst. Appl.* **2014**, *41*, 1937–1946. [[CrossRef](#)]
27. Al-Obeidat, F.; Al-Taani, A.T.; Belacel, N.; Feltrin, L.; Banerjee, N. A fuzzy decision tree for processing satellite images and landsat data. *Procedia Comput. Sci.* **2015**, *52*, 1192–1197. [[CrossRef](#)]
28. Zhou, G. Co-location Decision Tree for Enhancing Decision-Making of Pavement Maintenance and Rehabilitation. Ph.D. Thesis, Old Dominion University, Norfolk, VA, USA, 2011.
29. Zhan, D.; Hua, Z. Ensemble-based manifold learning for visualization. *J. Comput. Res. Dev.* **2005**, *42*, 1533–1537. [[CrossRef](#)]
30. Weinberger, K.; Saul, L. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vis.* **2006**, *7*, 77–90. [[CrossRef](#)]
31. Wang, J. Maximum Variance Unfolding. In *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 181–202.
32. Shao, J.D.; Rong, G. Nonlinear process monitoring based on maximum variance unfolding projections. *Expert Syst. Appl.* **2009**, *36*, 11332–11340. [[CrossRef](#)]
33. Liu, Y.J.; Chen, T.; Yao, Y. Nonlinear process monitoring and fault isolation using extended maximum variance unfolding. *J. Process Control* **2014**, *24*, 880–891. [[CrossRef](#)]
34. Ery, A.C.; Bruno, P. On the convergence of maximum variance unfolding. *J. Mach. Learn. Res.* **2013**, *14*, 1747–1770.
35. Vandenberghe, L.; Boyd, S.P. Semidefinite programming. *Soc. Ind. Appl. Math. Rev.* **1996**, *38*, 49–95. [[CrossRef](#)]
36. Weinberger, K.Q.; Packer, B.D.; Saul, L.K. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS-05), Bridgetown, Barbados, 6–8 January 2005; pp. 381–388.
37. Kardoulas, N.G.; Bird, A.C.; Lawan, A.I. Geometric correction of spot and Landsat imagery: A comparison of map- and GPS-derived control points. *Am. Soc. Photogramm. Remote Sens.* **1996**, *62*, 1171–1177.
38. Storey, J.C.; Choate, M.J. Landsat-5 Bumper-Mode geometric correction. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 2695–2703. [[CrossRef](#)]
39. Yang, W.J. The registration and mosaic of digital image remotely sensed. In Proceedings of the 11th Asian Conference on Remote Sensing, Guangzhou, China, 15–21 November 1990.
40. Kanazawaa, Y.; Kanatani, K. Image mosaicing by stratified matching. *Image Vision Comput.* **2004**, *22*, 93–103. [[CrossRef](#)]
41. Greiner, G.; Hormann, K. Efficient clipping of arbitrary polygons. *ACM Trans. Graph.* **1998**, *17*, 71–83. [[CrossRef](#)]
42. Liu, P.J.; Zhang, L.; Kurban, A. A method for monitoring soil water contents using satellite remote sensing. *J. Remote Sens.* **1997**, *1*, 135–139.
43. Qin, Z.; Karnieli, A. A mono-window algorithm for retrieving land surface temperature from Landsat TM data and its application to the Israel-Egypt border region. *Int. J. Remote Sens.* **2001**, *22*, 3719–3746. [[CrossRef](#)]
44. Mohammad, A.; Shi, Z.; Ahmad, Y. Application of GIS and remote sensing in soil degradation assessments in the Syrian coast. *J. Zhejiang Univ. (Agric. Life Sci.)* **2002**, *26*, 191–196.
45. Cliff, A.; Ord, J. Spatial Autocorrelation. *Environ. Plan.* **1973**, *7*, 725–734. [[CrossRef](#)]
46. Anselin, L. The local indicators of spatial association—LISA. *Geog. Anal.* **1995**, *27*, 93–115. [[CrossRef](#)]

47. Getis, A. Spatial autocorrelation. In *Handbook of Applied Spatial Analysis: Software Tools, Methods and Applications*; Fischer, M., Getis, A., Eds.; Springer: Berlin, Germany, 2010; pp. 255–278.
48. Zhao, S. Salt Marsh Classification and Extraction Based on HJ NDVI Time Series. Master's Thesis, Nanjing University, Nanjing, China, May 2015.
49. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
50. Van der Linden, S.; Rabe, A.; Held, M.; Jakimow, B.; Leitão, P.J.; Okujeni, A.; Schwieder, M.; Suess, S.; Hostert, P. The EnMAP-Box—A toolbox and application programming interface for EnMAP data Processing. *Remote Sens.* **2015**, *7*, 11249–11266. [[CrossRef](#)]



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).